NASA Technical Paper 2533

March 1986

NASA-TP-2533 19860013811

Interactive Algebraic Grid-Generation Technique

R. E. Smith
and M. R. Wiese

NASA

NASA Technical Paper 2533

1986

Interactive Algebraic Grid-Generation Technique

R. E. Smith
Langley Research Center
Hampton, Virginia

M. R. Wiese
Computer Sciences Corporation
Hampton, Virginia

National Aeronautics and Space Administration

Scientific and Technical Information Branch

Introduction

Finite-difference techniques are applied to solve field problems on grids which are ordered sets of discrete points in a coordinate system. Solution accuracy and computational expediency depend on the grid as well as the finite-difference technique. For instance, the accurate representation of boundary conditions and the resolution of solutions on regions of rapid change such as boundary layers, shocks, and separations in flow fields require a grid with particular characteristics. First, the grid should adapt to the physical boundaries so that boundary conditions can be readily applied. Second, the grid should be concentrated in regions of rapid change to accurately compute the solution there. Third, from a computational point of view it is desirable that the grid be uniform, the boundaries enclose a rectangle in two dimensions or a rectangular parallelepiped in three dimensions (fig. 1), and the exterior boundaries correspond to physical boundaries. If this is possible, overall computer program logic for the application of a finite-difference solution algorithm can be minimized and the process can be kept highly repetitive over the entire grid.

A direct (algebraic) approach to grid generation, for which an explicit functional relationship between the computational domain and the physical domain is known, has the advantage that changes to the grid are direct and are rapidly obtained. This report describes such an approach. It also describes the use of an interactive computer program for constructing a direct functional relationship between the computational domain and the arbitrary, simply connected, two-dimensional physical domains such that boundaries in the two domains map into each other. The technique is called the "two-boundary technique" and is described in general in references 1 to 4. The outlining features of the technique are derivatives of transfinite interpolation (ref. 3) and the technique can also be, in part, derived from the general multisurface equation (ref. 4). The two-boundary technique is best applied in an interactive environment. An interactive program based on the technique has been written, and its usage is described herein.

Background

For a large class of problems, an approach to grid generation which tends to satisfy both the accuracy and expediency requirements is to transform the governing equations from the original defining coordinate system (referred to as the physical-coordinate system) to a rectangular computational-coordinate system. The particular region of a coordinate system where a grid is defined is referred to as a domain. The computational domain can be expressed by

$$
\begin{aligned}
0 \leq \xi \leq 1 \\
0 \leq \eta \leq 1 \\
0 \leq \varsigma \leq 1
\end{aligned}
$$

where ξ, η, and ς are computational coordinates. (A list of symbols and abbreviations used in this paper appears after the references.) A transformation from the computational domain to the physical domain can be expressed as

$$
\left.
\begin{aligned}
x &= x(\xi, \eta, \varsigma) \\
y &= y(\xi, \eta, \varsigma) \\
z &= z(\xi, \eta, \varsigma)
\end{aligned}
\right\} \tag{1}
$$

where x, y, and z are physical coordinates. Similarly, a grid in one domain can be mapped into a grid in the other domain (fig. 2). When the transformation maps boundaries in the computational domain into boundaries in the physical domain, the term "boundary-fitted coordinate system" (ref. 1) is used to describe the transformation.

The transformation of the governing equations implies that derivatives with respect to the physical coordinates must be transformed to derivatives with respect to the computational coordinates. Reference 1 describes the transformation of the governing equations of fluid flow for which the chain rule of calculus is applied. The basic result of the transformation is inclusion of the Jacobian matrix

$$
\mathbf{J} = \begin{bmatrix}
\frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\
\frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\
\frac{\partial \varsigma}{\partial x} & \frac{\partial \varsigma}{\partial y} & \frac{\partial \varsigma}{\partial z}
\end{bmatrix} = \begin{bmatrix}
\frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \varsigma} \\
\frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \varsigma} \\
\frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \varsigma}
\end{bmatrix}^{-1}
$$

in the transformed equations of motion. Thus, for the application of finite-difference techniques, the primary function of grid generation is the determination of grid derivatives at grid node points. Normally, the process is to express the physical grid as a function of the computational grid and then to differentiate to obtain the inverse Jacobian matrix, which is itself inverted to produce the Jacobian matrix (ref. 1).

The application of the two-boundary technique can be thought of as two distinct processes. The first process is the definition of boundaries (referred to as the bottom, top, left, and right) and the functional description of the inscribed physical domain. The second process is the determination of the properly spaced grid in the physical domain which corresponds to a uniform grid in the computational domain.

A convenient way to represent the boundaries is to specify subsets of points and an interpolation

procedure to specify arbitrary boundary points. The choice of a boundary interpolation procedure can be dependent upon many things. For the interactive grid-generation program, it is assumed that the subset boundary definitions are densely specified and linear interpolation is sufficient for arbitrary boundary definition. Accuracy of the boundary representation is directly controlled by the number and location of the subsets of points. The essence of the grid-generation technique is connecting distributions of points on the bottom and top boundaries. Here simplicity and versatility are achieved by use of Hermite cubic polynomials as connecting functions. If only endpoints of the cubics are specified, the connecting functions are straight lines; however, if the position and the derivatives orthogonal to the bottom and top boundaries at the endpoints of the cubics are specified, an effective curved connecting function is defined (fig. 3). A distribution of points along the connecting function forms the initial grid. In the event that side boundaries to the domain are specified, the connecting curves between the bottom and top boundaries must be made to conform. Linear blending functions are applied to the initial grid to create a final grid which conforms to the side boundaries.

The second process, and the one most dependent upon interactive graphics, is the distribution of grid points along the bottom and top boundaries, the distribution of the magnitude of normal derivatives at the boundaries, and the distribution of grid points along the connecting function. A smooth-cubic-spline technique (ref. 5) has been adopted to control these distributions. The degree of smoothness is a function of user-specified parameters. The reason for using a smoothing process is that the versatility and continuity characteristics of cubic splines can be employed and the oscillation problem that can occur with unsmoothed cubic splines can be avoided. The control of the distribution of grid points is based on establishing a parametric variable such as approximated arc length along a curve (i.e., a boundary curve or a connecting curve), normalizing the parametric variable with respect to its maximum value, expressing each of the coordinates of the curve as a function of the parametric variable, and expressing the parametric variable as a function of a computational coordinate which is defined on the unit interval (i.e., $0 \leq \xi \leq 1$). The unit square is a control space (fig. 4) for the distributions along the curves. A discrete uniform distribution of the computational coordinate maps into an arbitrary distribution of the physical coordinates. Using interactive graphics to digitize a few points on the unit square for each control function allows for arbitrary control of the physical-grid distribution. For the two-boundary technique applied in two dimensions there are seven distributions to be determined. The primary purpose of the interactive program described herein is the determination of these control functions.

The Two-Boundary Technique

The two-boundary technique for grid generation is a methodology for establishing the mathematical expression (eqs. (1)) relating the computational domain to a physical domain. The methodology separates the boundary-grid definition from the interior-grid definition. The boundary grid is first defined, and then the interior grid is defined as a function of boundary position, boundary derivatives, and an independent variable t ($0 \leq t \leq 1$) spanning the two boundaries. Because of this approach, the algebraic expression of equations (1) in two dimensions is rewritten as

$$\left.\begin{aligned} x &= x(X_B(r), Y_B(r),\ X_T(s), Y_T(s),\ t) \\ y &= y(X_B(r), Y_B(r),\ X_T(s), Y_T(s),\ t) \end{aligned}\right\} \tag{2}$$

where $0 \leq r \leq 1$, $0 \leq s \leq 1$, $0 \leq t \leq 1$, $X_B(r), Y_B(r)$ are coordinates on the bottom boundary, and $X_T(s), Y_T(s)$ are coordinates on the top boundary. The parametric variables r, s, and t are expressed as functions of the computational coordinates.

The boundaries are described by ordered sets of points

$$\left\{X_B^i, Y_B^i\right\}_{i=1}^{i=N_B}$$

$$\left\{X_T^i, Y_T^i\right\}_{i=1}^{i=N_T}$$

where N_B and N_T are the numbers of defining points in each set. They are parameterized by computing the approximate arc lengths from the first points through all the points in the boundary sets. That is,

$$\begin{aligned} \bar{r}^i = \bar{r}^{i-1} + \Big[&(X_B^i - X_B^{i-1})^2 \\ &+ (Y_B^i - Y_B^{i-1})^2\Big]^{1/2} \qquad (\bar{r}^1 = 0; i = 2, 3, ..., N_B) \end{aligned}$$

$$\begin{aligned} \bar{s}^i = \bar{s}^{i-1} + \Big[&(X_T^i - X_T^{i-1})^2 \\ &+ (Y_T^i - Y_T^{i-1})^2\Big]^{1/2} \qquad (\bar{s}^1 = 0; i = 2, 3, ..., N_T) \end{aligned}$$

The approximate arc length corresponding to each point is normalized with respect to the maximum

value for each boundary by

$$r^i = \frac{\bar{r}^i}{\bar{r}^{N_B}} \quad (0 \le r \le 1)$$

$$s^i = \frac{\bar{s}^i}{\bar{s}^{N_T}} \quad (0 \le s \le 1)$$

and new sets of boundary points are described by

$$\Phi_1 \equiv \left\{r^i,\ X_B^i\right\}_{i=1}^{i=N_B} \qquad \Phi_2 \equiv \left\{r^i,\ Y_B^i\right\}_{i=1}^{i=N_B}$$

$$\Phi_3 \equiv \left\{s^j,\ X_T^j\right\}_{j=1}^{j=N_T} \qquad \Phi_4 \equiv \left\{s^j,\ Y_T^j\right\}_{j=1}^{j=N_T}$$

With the sets Φ_1, Φ_2, Φ_3, and Φ_4, arbitrary points and derivatives along the boundaries are described by

$X_B(r), Y_B(r)$	interpolation of r into Φ_1, Φ_2
$X_T(s), Y_T(s)$	interpolation of s into Φ_3, Φ_4
$\frac{dX_B(r)}{dr}, \frac{dY_B(r)}{dr}$	derivative interpolation of r into Φ_1, Φ_2
$\frac{dX_T(s)}{ds}, \frac{dY_T(s)}{ds}$	derivative interpolation of s into Φ_3, Φ_4

For the description herein all boundary interpolations are linear; however, in general the interpolation procedure along the boundaries is arbitrary.

The cubic function (refs. 1 and 2) connecting the two boundaries is

$$\left.\begin{aligned} x &= X_B(r)\,\alpha_1(t) + X_T(s)\,\alpha_2(t) + P(r)\frac{dY_B(r)}{dr}\alpha_3(t) \\ &\quad + Q(s)\frac{dY_T(s)}{ds}\alpha_4(t) \\ y &= Y_B(r)\,\alpha_1(t) + Y_T(s)\,\alpha_2(t) - P(r)\frac{dX_B(r)}{dr}\alpha_3(t) \\ &\quad - Q(s)\frac{dX_T(s)}{ds}\alpha_4(t) \end{aligned}\right\} \tag{3}$$

where

$$\left.\begin{aligned} \alpha_1(t) &= 2t^3 - 3t^2 + 1 \\ \alpha_2(t) &= -2t^3 + 3t^2 \\ \alpha_3(t) &= t^3 - 2t^2 + t \\ \alpha_4(t) &= t^3 - t^2 \end{aligned}\right\} \quad (0 \le t \le 1)$$

$P(r)$ is the magnitude of the normal derivative along the bottom boundary, and $Q(s)$ is the magnitude of the normal derivative along the top boundary.

The cubic curve in equations (3) is specified by: (1) the coordinates on the bottom and top boundaries which are functions of approximate-arc-length variables, (2) the derivatives of the bottom and top coordinates with respect to the approximate-arc-length variables, and (3) the magnitudes of the derivatives at the bottom and top points. The blending functions $\alpha_1(t)$, $\alpha_2(t)$, $\alpha_3(t)$, and $\alpha_4(t)$ are shown in figure 5.

The magnitudes of the derivatives affect the orthogonality of the cubics relative to the boundary curves and are expressed as functions of the approximate arc lengths. Increasing the magnitudes extends the orthogonality of the cubic away from the boundary curves, and excessively large values of the magnitudes can cause the cubics to intersect themselves between the boundaries, which is unacceptable for a grid.

The approximate arc lengths along the bottom and top boundaries are expressed as a function of the computational coordinate ξ. There are four distribution functions that determine the spacing of the cubics. They are

$$\left.\begin{aligned} r &= f_1(\xi) \\ s &= f_2(\xi) \\ P(r) &= K_B\, f_3(\xi) \\ Q(s) &= K_T\, f_4(\xi) \end{aligned}\right\} \tag{4}$$

The functions $f_1(\xi)$, $f_2(\xi)$, $f_3(\xi)$, and $f_4(\xi)$ are called control functions, and their determination is the second phase of the grid-generation process. The constants K_B and K_T scale the magnitudes of the derivatives at the bottom and top boundaries. Figure 6 depicts the process for determining arbitrary boundary points and derivatives from the initial tabular descriptions and the connecting function.

A distribution of the variable t between 0 and 1 specifies a distribution of points along the curve for which the first and last points are at the top and bottom boundaries (fig. 3). The variable t could be made a direct function of the computational coordinate η, but instead it is made a function of the normalized approximate arc length along each cubic and then the approximate arc length is made a function of η. This step requires that equations (3) be used twice to compute a grid and is included because the application of side boundaries, to be discussed subsequently, is dependent on an arc-length parameterization. The variable t is made an empirical function of normalized approximate arc length by inversely computing the approximate arc length as a function of t at uniform steps in equations (3). Values of t versus normalized approximate arc length are determined from

$$t_j = \frac{j-1}{N_P - 1} \quad (j = 1, 2, ..., N_P)$$

$$\bar{u}_j = \bar{u}_{j-1} + \left[(x_j - x_{j-1}^2) + (y_j - y_{j-1}^2)^{1/2}\right] \quad (\bar{u}_1 = 0; j = 2, 3, ..., N_P)$$

and

$$u_j = \frac{\bar{u}_j}{\bar{u}_{N_P}} \quad (0 \leq u_j \leq 1)$$

where x_j and y_j are computed from equations (3). For each cubic there is a set

$$\Psi \equiv \{u_j, t_j\}_{j=1}^{j=N_P}$$

The number of points N_P along the cubic connecting function is chosen to yield a sufficiently accurate approximation of the arc length. Since u and t are monotonically increasing functions of each other, the values of Ψ can be interpolated for a distribution of the variable t corresponding to a distribution of the approximate arc length. The normalized approximate arc length is expressed as a function of the computational coordinate η by $u = f_5(\eta)$.

Given the boundary data, there are now five control functions that are necessary to map a uniform computational grid into a physical grid by use of the two-boundary technique. Thus, for

$$\xi(I) = \frac{I-1}{N-1} \quad (I = 1, 2, ..., N)$$

$$\eta(J) = \frac{J-1}{M-1} \quad (J = 1, 2, ..., M)$$

which is a uniform distribution of the computational domain, where N is the number of grid points in the ξ direction and M is the number of grid points in the η direction. Distributions of normalized approximate-arc-length parametric variables,

$$\{r_I\}_{I=1}^{I=N}$$

$$\{s_I\}_{I=1}^{I=N}$$

$$\{u_j\}_{J=1}^{J=M}$$

are computed with the functions $f_1(\xi)$, $f_2(\xi)$, and $f_5(\eta)$, and the distribution

$$[t(I,J)]_{\substack{J=1\\I=1}}^{\substack{J=M\\I=N}}$$

is computed with the functions $f_3(\xi)$, $f_4(\xi)$, and equations (3). Finally, again through use of equations (3), the grid

$$\{x(I,J), y(I,J)\}_{\substack{J=1\\I=1}}^{\substack{J=M\\I=N}}$$

is computed. Figure 7 depicts the process of computing a grid with equations (3) and the boundary data.

At this point, $\{x(1,J), y(1,J)\}_{J=1}^{J=M}$ and $\{x(N,J), y(N,J)\}_{J=1}^{J=M}$ define the left- and right-boundary grid points. However, prespecified left- and right-boundary curves are defined by the sets

$$\{X_L^k, Y_L^k\}_{k=1}^{k=N_L}$$

$$\{X_R^k, Y_R^k\}_{k=1}^{k=N_R}$$

where N_L and N_R are the numbers of defining points in each set. The side boundaries are parameterized by computation of the approximate arc lengths from the first points through all the points in the boundary sets. That is,

$$\bar{v}^k = \bar{v}^{k-1} + \left[(X_L^k - X_L^{k-1})^2 + (Y_L^k - Y_L^{k-1})^2\right]^{1/2} \quad (\bar{v}^1 = 0; k = 2, 3, ..., N_L)$$

$$\bar{w}^k = \bar{w}^{k-1} + \left[(X_R^k - X_R^{k-1})^2 + (Y_R^k - Y_R^{k-1})^2\right]^{1/2} \quad (\bar{w}^1 = 0; k = 2, 3, ..., N_R)$$

The approximate arc length corresponding to each point is normalized with respect to the maximum value for each boundary by

$$v^k = \frac{\bar{v}^k}{\bar{v}^{N_L}} \quad (k = 1, 2, ..., N_L)$$

$$w^k = \frac{\bar{w}^k}{\bar{w}^{N_R}} \quad (k = 1, 2, ..., N_R)$$

and the left and right boundaries are described by

$$\Theta_1 = \left\{v^k, X_L^k\right\}_{k=1}^{k=N_L} \qquad \Theta_2 = \left\{v^k, Y_L^k\right\}_{k=1}^{k=N_L}$$

$$\Theta_3 = \left\{w^k, X_R^k\right\}_{k=1}^{k=N_R} \qquad \Theta_4 = \left\{w^k, Y_R^k\right\}_{k=1}^{k=N_R}$$

Grid coordinates for the left and right boundaries are obtained by interpolating $u(J) = f_5(\eta(J))$ into the sets Θ_1, Θ_2, Θ_3, and Θ_4 such that

$$x_L(J) = \text{Interpolation of } u(J) \text{ into } \Theta_1 \equiv \{v^k, X_L^k\}_{k=1}^{k=N_L}$$

$$y_L(J) = \text{Interpolation of } u(J) \text{ into } \Theta_2 \equiv \{v^k, Y_L^k\}_{k=1}^{k=N_L}$$

$$x_R(J) = \text{Interpolation of } u(J) \text{ into } \Theta_3 \equiv \{w^k, X_R^k\}_{k=1}^{k=N_R}$$

$$y_R(J) = \text{Interpolation of } u(J) \text{ into } \Theta_4 \equiv \{w^k, Y_R^k\}_{k=1}^{k=N_R}$$

The final grid that conforms to the specified left and right boundaries is obtained by blending, where

$$\left.\begin{aligned} x(I,J) &= x(I,J) + [x_L(J) - x(1,J)]\{1 - f_6[\xi(I)]\} \\ &\quad + [x_R(J) - x(N,J)]\{f_7[\xi(I)]\} \\ y(I,J) &= y(I,J) + [y_L(J) - y(1,J)]\{1 - f_6[\xi(I)]\} \\ &\quad + [y_R(J) - y(N,J)]\{f_7[\xi(I)]\} \end{aligned}\right\} \tag{5}$$

where $I = 1, 2, ..., N$ and $J = 1, 2, ..., M$. The control functions $f_6[\xi(I)]$ and $f_7[\xi(I)]$ specify the blending of the side boundaries with the grid that is obtained through use of the bottom and top boundaries (fig. 8).

Grid Spacing Control

The second phase of the algebraic grid-generation technique is the control of the spacing of grid points. For uniform distributions of ξ and η, the functions $f_1(\xi)$, $f_2(\xi)$, $f_3(\xi)$, $f_4(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$ along with the constants K_B and K_T determine distributions of the parametric variables r, s, and u and the derivative magnitudes $P(r)$ and $Q(s)$. The functions used to control the spacing of grid points can be analytical. For example, the function

$$f_5(\eta) = \frac{e^{K\eta} - 1}{e^K - 1}$$

would concentrate grid points close to the bottom or the top boundary depending on the magnitude and the sign of the constant K.

For a general grid-generation procedure, greater flexibility in the control of grid spacing is needed than can be obtained from specific analytical formulas. A general approach is the application of cubic-spline functions constrained to be inside the unit square. The essence of the control domain (unit square, see fig. 4) is that the abscissa corresponds to the percentage of grid points and the ordinate corresponds to a particular control function which, in turn, relates to the geometric definition of the physical domain. A control function can be specified by digitizing a few points in the unit square and then applying the spline continuity conditions. The functions $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$ pass through the origin and monotonically increase to the point (1,1). The functions $f_3(\xi)$ and $f_4(\xi)$ must be single valued but otherwise are free to be anywhere in the unit interval. The derivative magnitudes $P(r)$ and $Q(r)$ are obtained by multiplying the constants K_B and K_T by $f_3(\xi)$ and $f_4(\xi)$, respectively.

The difficulty with using cubic splines for control is that oscillations can be inadvertently introduced into the control functions. This problem is avoided by using a smoothing cubic-spline technique and specifying the amount of smoothing as well as the digitized points in the unit square. The technique that is used is described in detail in reference 6 and is described in less detail in reference 5. The features of the application of smooth cubic splines are outlined below.

For grid spacing control, the independent variable $\phi(0 \leq \phi \leq 1)$ represents a computational coordinate and the variable $\theta(0 \leq \theta \leq 1)$ represents a control function. A discrete set of points

$$A \equiv \{\phi_i, \theta_i\}_{i=1}^{i=N_C}$$

specifies a cubic spline. The conditions on the points for the functions $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$ are

$$0 \leq \theta_2 \leq \theta_3 \leq ... \leq 1$$

$$0 \leq \phi_2 \leq \phi_3 \leq ... \leq 1$$

For the functions $f_3(\xi)$ and $f_4(\xi)$ the conditions are

$$0 \leq \theta_2 \leq \theta_3 \leq ... \leq 1$$

$$0 \leq \phi_2 \leq \phi_3 \leq ... \leq 1$$

A cubic spline in the (ϕ, θ) plane with the knots $\{\phi_i, \theta_i\}_{i=1}^{i=N_C}$ is

$$\begin{aligned} F(\phi) &= \{\mu(\phi_i)\}_{i=1}^{i=N_C-1} \\ &= \{a_i + b_i(\phi - \phi_i) + c_i(\phi - \phi_i)^2 \\ &\quad + d_i(\phi - \phi_i)^3\}_{i=1}^{i=N_C-1} \end{aligned}$$

where

$$\left.\begin{aligned} \mu_i(\phi_{i+1}) &= \mu_{i+1}(\phi_{i+1}) \\ \frac{d\mu_i}{d\phi}(\phi_{i+1}) &= \frac{d\mu_{i+1}}{d\phi}(\phi_{i+1}) \\ \frac{d^2\mu_i}{d\phi^2}(\phi_{i+1}) &= \frac{d^2\mu_{i+1}}{d\phi^2}(\phi_{i+1}) \end{aligned}\right\} \quad (i = 1, 2, ..., N_C - 2; 0 \leq \phi_i \leq 1)$$

The coefficients $\{a_i, b_i, c_i, d_i\}_{i=1}^{i=N_C-1}$ are undetermined parameters whose solutions define $F(\phi)$. The objective is to find the coefficients which minimize the integral of the second derivative squared

$$\int_0^1 [F''(\phi)]^2 \, d\phi$$

subject to the constraint

$$\sum_{i=1}^{N_C} \left[\frac{F_i(\phi) - \theta_i}{\delta\theta_i}\right] \leq C$$

where C is a positive constant specifying the extent of smoothing and $\delta\theta_i$ is the allowed deviation of the spline function from the ith ordinate θ_i. The restated objective is to find the smoothest cubic spline passing within the bounds

$$\theta_i - \delta\theta_i \leq \theta_i \leq \theta_i + \delta\theta_i$$

$$B \equiv \{\delta\theta_i\}_{i=1}^{i=N_C}$$

$$\delta\theta_1 = \delta\theta_{N_C} = 0$$

where $\delta\theta_i$ represents the maximum derivations of the spline function from the specified ordinates θ_i (fig. 9).

The method of Lagrange multipliers from the calculus of variations is used to find the parameters

$$\{a_i, b_i, c_i, d_i\}_{i=1}^{i=N_C-1}$$

The solution algorithm for spline smoothing can be found in reference 6. The techniques exist in subroutine form, where the sets A and B and the constant C are input.

The set A is obtained by digitizing points on the unit plane, the constant C is set equal to $N_C - 1$, and the set B is obtained from the linear relation

$$\delta\theta_i = m_0 + m_1 \left(\frac{\theta_{i+1} - \theta_{i-1}}{\phi_{i+1} - \phi_{i-1}} \right) \quad (i = 2, 3, ..., N_C - 1) \tag{6}$$

Using this relationship allows rapid choice of the deviations which prescribe the amount of smoothing. Choosing m_0 and m_1 to be zero results in $B = 0$, and the control function $F(\phi)$ is a cubic spline fit to the set A. Cubic splines are subject to oscillations which are easily observed by plotting the derivative functions. Choosing m_0 and m_1 to be large results in large allowable deviations in the set B, and the control function $F(\phi)$ is a straight line. The objective is to allow just enough dispersion in set B to achieve a smooth control function which satisfies the conditions described above and produces the desired grid concentrations. This process is ideally suited for interactive computer graphics, where the set A can be digitized with a cursor, the constants m_0 and m_1 are alphanumeric input, and the results can be rapidly evaluated.

Interactive Algebraic Grid Generation

A computer program based on the algebraic technique described above is computationally simple because only boundary coordinates and control points on the unit square are required input. Linear interpolation, the connecting function, and the spline smoothing technique are the mathematical components. An interactive program, however, is more complex than a noninteractive program because the user and the program must communicate through questions, responses, and graphical displays. Also, fault tolerances must be coded so that interactive input errors will not cause the program to abort.

The program described herein and called TBGG (two-boundary grid generation) is coded in FORTRAN V and, in its present form, runs on the Prime 750 computer and the Control Data CYBER 170 series computers. There are two versions of the program—one for the Precision Visuals, Inc., DI-3000 graphics library (ref. 7) and one for the Tektronix, Inc., PLOT-10 graphics library (ref. 8). The DI-3000 version requires 231 000 words of memory and the PLOT-10 version requires 170 600 words of memory. With minor modifications, the program should run on other computers similar to the ones mentioned above. This section describes the program input and interactive use.

There are two forms of noninteractive input to the program. In the first form, the boundary data are presented to the program in an input file (DATANEW, see table I) and a grid based on linear control functions and no orthogonality is displayed with options for interactive development of the grid spacing. The second form of data input is a restart file (RESTART) created at the end of a previous interactive session. The restart file allows the continuation of the interactive development of the grid spacing. The output from the program is a free-formatted file called GRIDOUT, which contains the grid coordinates, and an unformatted file called RESTART, which contains the parameters and data for the last interactively developed grid.

Program Usage

Before operating program TBGG to generate a grid, a file called DATANEW, which contains the problem title, the number of points on each boundary, and the x- and y-coordinates of each boundary point, must be created. The file DATANEW is read by the program when "NEW" followed by a carriage return (CR) is the interactive response at the initiation of the program. If "OLD" followed by a CR is the interactive response, the file RESTART (generated in a previous session) is read. Examples and their associated DATANEW files are described in the appendix.

The output file RESTART is an unformatted file and contains the grid spacing information developed during an interactive session. The objective of RESTART is to recover exactly the conditions that existed in a previous interactive session. The file cannot be edited, and the specific format and order of the data are not described herein.

After the DATANEW file or the RESTART file has been read, a grid and a set of options are presented on the terminal screen. The nature of the first grid display depends on whether DATANEW or RESTART is used for input. If DATANEW is used, uniform distributions are assumed for $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$. The orthogonality constants K_B and K_T are set to zero. If side boundaries are not specified or if they are straight lines, the curves connecting the bottom and top boundaries are straight lines and the spacing between grid points is uniform. If RESTART is used, the grid is based on the values $f_1(\xi)$, $f_2(\xi)$, $f_3(\xi)$, $f_4(\xi)$, $f_5(\eta)$, $f_6(\xi)$, $f_7(\xi)$, K_B, and K_T previously developed.

General Options

The general options are displayed with the grid (fig. 10), and they are briefly described in table II. An option is invoked by typing the option number followed by a CR.

Boundary and Connecting Function Control Distributions (General Options 1, 2, 3, and 4)

General options 1, 2, 3, and 4 are used to specify the distribution of grid points along the bottom and top boundaries, the concentration effect relative to the side boundaries, and the distribution of grid points along the connecting function. These options are for the development of the control functions $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$. Typing 1, 2, 3, or 4 followed by a CR will cause a display of local options relative to each control function. A display of local options is shown in figure 11. The first three local options define control points within the unit square by use of programmed analytical functions with the objective of accelerating the process of creating relatively simple control functions. Local option 1 creates a linear distribution of points, local option 2 creates an exponential distribution of points, and local option 3 creates a distribution from a combination of two exponential functions. One user-defined parameter is required for local option 2 and two parameters are required for local option 3. Local option 4 enables the user to digitize points within the unit square, thereby defining arbitrary control functions. Local option 5 applies only to the bottom and top boundaries. It automatically sets the top-boundary distribution equal to the bottom-boundary distribution, or vice versa.

Local option 2 generates an exponential distribution of points based on the function

$$\bar{\tau} = \frac{e^{K\tau} - 1}{e^{K} - 1} \quad (0 \leq \tau \leq 1; 0 \leq \bar{\tau} \leq 1)$$

Positive values of the parameter K produce relatively small first derivative magnitudes of the control function near the point (0,0), and negative values of the parameter K produce relatively small first derivative magnitudes of the control function near the point (1,1) (fig. 12). The variables τ and $\bar{\tau}$ are used locally to represent the independent and dependent variables for either of the exponential options. For the purposes of the program the range of the parameter K is $0.001 \leq K \leq 50$.

Local option 3 uses a combination of two exponentials to generate nominal control points on the unit square. The equations are

$$\bar{\tau} = K_1 \left[\frac{e^{(K_2/K_1)\tau} - 1}{e^{K_2} - 1}\right] \quad (0 \leq \tau \leq K_1; 0 \leq \bar{\tau} \leq K_1)$$

$$\bar{\tau} = K_1 + (1 + K_1)\left[\frac{e^{-K_2(\tau - K_1)/(1-K_1)}}{e^{-K_2} - 1}\right]$$

$$(0 < K_1 \leq 1; 0.001 \leq K_2 \leq 50)$$

Increasingly positive values of K_2 cause decreasing first derivative magnitudes of the control function near the point (0,0) and the point (1,1) in the unit square. An inflection occurs at the point with coordinates (K_1, K_1), where there will be a relatively large first derivative magnitude. Conversely, increasingly negative values of K_2 cause increasing first derivative magnitudes of the control function near points (0,0) and (1,1); near the inflection point with coordinates (K_1, K_1) the first derivative will be relatively small. Figure 13 demonstrates this option.

After typing local option numbers 2 or 3 followed by a CR, the range of the parameter or parameters is displayed and the user must respond by typing a value or values separated by a comma and followed by a CR.

Arbitrary Control Functions

As previously stated, local option 4 allows the user to specify points within the unit square for control function definition. For the five control functions $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$, the first point is always (0,0) and the last point is always (1,1). These points are preset by the program. Only the points in between are specified by the user, and they must be monotonically increasing. The control functions $f_3(\xi)$ and $f_4(\xi)$ have different restrictions and are discussed with the orthogonality option (general option 5). Typing 4 followed by a CR displays the menu and the unit square shown in figure 14. Also, the digitizing cursor is activated and appears on the screen. The digitizing process is the movement of the cursor

to place it at desired points inside the unit square along with input from the keyboard to signal the program to accept a point. In order to implement the process, a set of specified commands (keyboard characters) must be used, and they are listed on the display (fig. 14). The commands are used to inform the program on what the user wants done. A description of the commands is presented in table III to acquaint the user with their purpose. The control of the placement of the cursor is accomplished through the use of analog devices on the terminal (fig. 14). Note that when first digitizing a control function, the F function is used to indicate to the program the point at abscissa of 1 and the S function indicates to the program to proceed to the next step.

Visual Aid for Digitizing Boundary Control Functions (P Function)

The P digitizing function is a visual aid for establishing in the unit square points which can be used to define the control functions $f_1(\xi)$ and $f_2(\xi)$ through use of general options 1 and 2. The P function displays the boundary curves and draws dotted connecting grid curves for the number of curves specified at the beginning of the interactive session or specified with general option 8. It is best to use a small number of grid curves to avoid visual confusion. This interactive interchange generates points in the unit square based on where the user thinks grid curves should intersect the boundary curves.

The option P is invoked by typing the character P while digitizing control function $f_1(\xi)$ for the bottom boundary and $f_2(\xi)$ for the top boundary (general options 1 and 2). This instruction causes the display shown in figure 15 to appear on the terminal screen. The user is asked which grid line is to be placed at a given arc length. The range of grid lines is also displayed. The number is typed and followed by a CR. The cursor appears on the screen and the user places it at the desired point as close as possible to the boundary curve. If the grid curve has not been used before, the character I is typed to indicate to the program that a new point is being digitized. If this is a modification to a previously used grid curve, the character C is typed to indicate to the program that a change to an existing control point is being made. Note that a CR does not follow the I or C commands. The display of the unit square and associated options appears on the screen with the "loaded point" ($\delta\theta_i = 0$, described in eqs. (6)) corresponding to what has been digitized. The abscissa of the point is $I/(N-1)$, where I corresponds to the Ith grid curve and N is the total number of grid curves. The ordinate is the normalized approximate arc length along the boundary curve nearest the intersection of the cross lines. The point is also denoted in the option P display for later reference.

Boundary Distribution Interchange

A fifth local option is provided for the bottom- and top-boundary-grid distributions. This option sets the control function for the bottom boundary equal to the control function for the top boundary, or vice versa. This option is invoked by typing 5 followed by a CR when it is provided on the menu.

Control Function Smoothing and Display

After the control points have been chosen, a display of the smooth spline fit to the control points for the number of grid points specified at the beginning of the program or in the file RESTART is presented on the terminal screen. Three local options relative to the spline fit are also presented. An example of this display is shown in figure 16. This display allows the user to ensure that: (1) the control function has the desired characteristics; (2) the control function, the first derivative, and the second derivative are sufficiently continuous; and (3) the control function is monotonically increasing for $f_1(\xi)$, $f_2(\xi)$, $f_5(\eta)$, $f_6(\xi)$, and $f_7(\xi)$. The spline fit options are: (1) change the smoothing parameters m_0 and m_1 in equations (6), (2) generate a different control function, or (3) accept the distribution and continue to display the resultant grid and general options. The parameters m_0 and m_1 are initially defined in the program to be 0.01.

Grid Orthogonality (General Option 5)

The two-boundary technique allows control of grid orthogonality through the definition of the functions $P(r)$ and $Q(s)$ in equations (4), where

$$P(r) = K_B\, f_3(\xi)$$

$$Q(s) = K_T\, f_4(\xi)$$

Unlike the previously described grid spacing control functions, $f_3(\xi)$ and $f_4(\xi)$ do not have to be monotonically increasing in the control domain between the origin (0,0) and the point (1,1).

General option 5 is invoked by typing 5 followed by a CR when the general options display is on the terminal screen. The option causes the display shown in figure 17 to appear on the terminal screen. The values K_B and K_T are input from the keyboard as alphanumeric constants within the range shown in the display. The values are separated by a comma and followed by a CR. Three local options are shown

in the display. The options are: (0) constant distributions of $f_3(\xi)$ and $f_4(\xi)$, (1) user-defined distributions of $f_3(\xi)$ and $f_4(\xi)$, or (2) keep current distributions. The options are input for both the bottom boundary and the top boundary, and the two values are separated by a comma and followed by a CR. If local options 0 or 2 are invoked, the grid is recomputed and the primary display showing the general options and grid appears on the terminal screen. If local option 1 is invoked, the display described above for arbitrary distributions appears on the terminal screen. The difference between digitizing control points for this option and those previously described is that the ordinate of the first point is digitized and the abscissa is set equal to 0. When the digitizing option F is executed, the ordinate of the last point is digitized and the abscissa is set equal to 1. The digitizing option S indicates to the program to spline smooth the control points and to display the smooth control function and smoothing options. The digitizing process caused by local option 1 is displayed.

After the control functions $f_3(\xi)$ and $f_4(\xi)$ have been determined, the grid is recomputed and the grid and general options are displayed. Note again that the orthogonality functions $f_3(\xi)$ and $f_4(\xi)$ are arbitrarily specified according to where the user wants the relative effect of the orthogonality to be along the bottom and top boundaries. The constants K_B and K_T govern the magnitude of the orthogonality.

Plotting of Grid Derivatives (General Option 6)

An extremely important aspect of an interactive grid-generation procedure is the ability to visually assess the quality of the grid. In addition to inspection of the grid itself, visual inspection of the derivatives of the physical grid coordinates with respect to the computational coordinates is easily made available to the interactive user of program TBGG.

In program TBGG, general option 6 provides for the plotting of the scaled grid derivatives along any grid curve. General option 6 is invoked by typing 6 followed by a CR when the general options display shown in figure 10 is presented on the terminal screen. If this instruction is used, the additional display shown in figure 18 also appears in the general options display. The additional display asks the user to enter the line type (ξ or η coordinate) to be differentiated, the coordinate with respect to which the derivative is being taken (1 for x and 2 for y), the coordinate direction in which the derivative is to be plotted (1 for $\xi(I)$ and 2 for $\eta(J)$), and the index number of the curve along which the derivative is taken. The numbers are separated by commas and followed by a CR. An example is the derivative of the y-coordinate with respect to the ξ-coordinate along the 10th curve in the η-direction. The interactive response is 2,1,2,10 CR. A display similar to the one shown at the bottom of figure 18 appears on the terminal screen. The derivative is plotted normalized relative to the lengths of the display axes, and the stretch factor is displayed under the plot. The response 0,0,0,0 CR indicates to the program to end the derivative display and causes the general option display to appear on the terminal screen.

Grid Enlargement (General Option 7)

In order to see the detailed characteristics of a grid, it is necessary to plot the grid at a sufficiently large scale. On a small screen it is not possible to enlarge the entire grid, but a section of the grid can be arbitrarily enlarged. General option 7 provides for the enlargement of a section of the grid that is plotted on the general options display (fig. 10).

General option 7 is invoked by typing 7 followed by a CR. The user is asked the multiple of enlargement that is desired. This number is typed followed by a CR. The digitizing cross lines appear on the terminal screen with the present grid. The user places the intersection of the cross lines at the center of the region that is to be enlarged and types the character I. The enlarged grid (fig. 19) appears on the terminal screen with the instructions to continue the enlargement or to return to the general options. Note that a CR does not follow the character I.

Changing the Number of Grid Points (General Option 8)

A sparse grid can be used in the early stages of the development of the control functions and orthogonality parameters that determine the spacing of grid points. Using a sparse grid saves time both in the computing of the grid and in the drawing of the displays, and the amount of information presented to the user is not overwhelming. Once the general characteristics have been determined with use of a sparse grid, the number of grid points can be set to any desired value. However, to keep the memory size of program TBGG within reasonable limits, the present maximum grid size is 100×100 (100 points in the ξ-direction and 100 points in the η-direction). This can be changed in the program parameter statements. The minimum grid size is 4×4.

General option 8 is invoked by typing 8 followed by a CR when the general options and grid (fig. 10) are displayed on the terminal screen. A question will appear on the terminal screen asking how many grid points are desired. The user responds by typing the number of desired points in the ξ-direction,

a comma, the number of points in the η-direction, and a CR. The new grid is computed and displayed on the terminal screen with the general options.

Terminating the Program (General Option 9)

General option 9 terminates the program. It also writes the files GRIDOUT and RESTART. Note that GRIDOUT and RESTART are local files, and the user must actively store them if they are to be permanently saved.

Conclusions

An algebraic grid-generation technique that is easy to understand, easy to apply, and has a high degree of generality has been presented. Also, an interactive computer program for applying the technique in two dimensions has been presented. Boundary definitions are provided as ordered sets of points, and the spacing of a grid within the boundaries is governed by a set of control functions which are developed interactively. The control functions are based on an adoption of smooth-cubic-spline functions defined for a control space, which is the unit square. The process allows complete user control of the Hermite interpolation and linear blending. The result is an ordered set of points in a physical-coordinate system which corresponds to a uniform grid in a rectangular computational-coordinate system. Grid derivatives that are used in the solution of partial differential systems can be obtained by numerical differentiation of the physical grid with respect to the computational coordinates and the computation of inverse relations.

The interactive program not only provides a means of quickly creating grids, but it also provides the opportunity for the user to quickly evaluate and draw conclusions about the suitability of grids for particular applications. The program is user friendly, with prompts for each step of the process, and the program tolerates a variety of user errors.

NASA Langley Research Center
Hampton, VA 23665-5225
November 12, 1985

Appendix

Examples

The first example is the creation of a C-type grid about an airfoil. The boundary data for the DATANEW file are shown in table IV. The grid, the control functions, and the constants K_B and K_T are shown in figure 20.

The second example is an O-type grid about the same airfoil presented above. In this case the bottom-boundary data are the same as those in table IV except that they are defined clockwise. The "signs" of the orthogonality constants K_B and K_T are opposite to those in the first example. The top boundary is circular and the left and right boundaries are identical, extending from the airfoil to the outer boundary. Table V shows the DATANEW file for this example. Figure 21 shows the grid, the control functions, and the constants K_B and K_T.

The third example is a nozzle configuration in which no left or right boundaries are specified. The boundary data are shown in table VI, and the grid, the control functions, and the constants K_B and K_T are shown in figure 22. This example is illustrative of orthogonality. If the same magnitude of orthogonality is used in the center of the nozzle as near the ends, the grid would overlap. The functions $f_3(\xi)$ and $f_4(\xi)$ both have a value of one at the ends and approach zero at the center, which allows the grid to have acceptable characteristics.

The fourth example has the same bottom and top boundaries as the third example. Side boundaries have been added to show their effect. The functions $f_6(\xi)$ and $f_7(\xi)$ control how far into the grid the side boundaries modify the initial grid. Table VII shows the DATANEW file and figure 23 shows the grid, the control functions, and the constants K_B and K_T.

References

1. Smith, R. E.: *Two-Boundary Grid Generation for the Solution of the Three-Dimensional Compressible Navier-Stokes Equations.* NASA TM-83123, 1981.
2. Smith, Robert E.; and Weigel, Barbara L.: Analytic and Approximate Boundary-Fitted Coordinate Systems for Fluid Flow Simulation. AIAA-80-0192, Jan. 1980.
3. Gordon, William J.; and Thiel, Linda C.: Transfinite Mappings and Their Application to Grid Generation. *Numerical Grid Generation*, Joe F. Thompson, ed., Elsevier Science Publ., Co., Inc., c.1982, pp. 171–192.
4. Eiseman, Peter R.; and Smith, Robert E.: Mesh Generation Using Algebraic Techniques. *Numerical Grid Generation Techniques*, NASA CP-2166, 1981, pp. 73–120.
5. Smith, R. E.; Kudlinski, R. A.; and Everton, E. L.: A Grid Spacing Control Technique for Algebraic Grid Generation Methods. AIAA-82-0226, Jan. 1982.
6. Reinsch, C. H.: Smoothing by Spline Functions. *Numer. Math.*, vol. 10, no. 3, 1967, pp. 177–183.
7. *DI-3000 User's Guide.* Doc. G-5, Central Scientific Computing Complex, Precision Visuals, Inc., Mar. 1984.
8. *Plot-10 Terminal Control System—User's Manual.* Doc. No. 062-1474-00, Tektronix, Inc., c.1974. (Reprinted Dec. 1976.)

Symbols and Abbreviations

A — set description of points for smoothing spline functions

a, b, c, d — coefficients for smoothed cubic splines

B — set of deviations

C — constant specifying extent of smoothing

CR — carriage return

$\frac{dX_B(r)}{dr}, \frac{dY_B(r)}{dr}$ — derivatives of x- and y-coordinates with respect to r interpolated along bottom boundary

$\frac{dX_T(s)}{ds}, \frac{dY_T(s)}{ds}$ — derivatives of x- and y-coordinates with respect to s interpolated along top boundary

$\frac{d\mu(\phi)}{d\phi}$ — first derivative of cubic-spline representation

$\frac{d^2\mu(\phi)}{d\phi^2}$ — second derivative of cubic-spline representation

F — functional representation of smoothed cubic splines

$f_1(\xi), f_2(\xi)$ — control functions for distribution of grid points along bottom and top boundaries

$f_3(\xi), f_4(\xi)$ — control functions for relative magnitude of normal derivative at bottom and top boundaries

$f_5(\eta)$ — control function for distribution of grid points along connecting function

$f_6(\xi), f_7(\xi)$ — control functions for relative effect of left and right side boundaries

I, J — grid point indices in ξ and η directions

i — index for boundary point description

$\mathbf{J}$ — Jacobian matrix,

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} & \frac{\partial \xi}{\partial z} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} & \frac{\partial \eta}{\partial z} \\ \frac{\partial \varsigma}{\partial x} & \frac{\partial \varsigma}{\partial y} & \frac{\partial \varsigma}{\partial z} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} & \frac{\partial x}{\partial \varsigma} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} & \frac{\partial y}{\partial \varsigma} \\ \frac{\partial z}{\partial \xi} & \frac{\partial z}{\partial \eta} & \frac{\partial z}{\partial \varsigma} \end{bmatrix}^{-1}$$

j — index for points along the connecting function

K — concentration parameter

K_1, K_2 — control function parameters for bi-exponential function

K_B, K_T — constants for magnitude of normal derivative at bottom and top boundaries (KB and KT in computer displays)

k — index denoting point number for left- and right-boundary descriptions

M — total number of points along η boundary

m_0, m_1 — parameters to compute deviations (M0 and M1 in computer displays)

N — total number of points along ξ boundary

N_B, N_T — number of points describing bottom and top boundaries

N_L, N_R — number of points describing left and right boundaries

N_P — number of points used to accurately compute arc length along connecting function

$P(r), Q(s)$ — magnitudes of normal derivatives at bottom and top boundaries

r, s — normalized approximate arc lengths along the bottom and top boundaries

$\bar{r}, \bar{s}$ — approximate arc lengths along the bottom and top boundaries

t — parametric variable in Hermite's interpolation functions

u — normalized approximate arc length along connecting function

$\bar{u}$ — approximate arc length along connecting function

v, w — normalized approximate arc lengths along left and right boundaries

$\bar{v}, \bar{w}$ — approximate arc lengths along left and right boundaries

$X(I,J)$, $Y(I,J)$ — grid coordinates

$X_B(r), Y_B(r)$ — interpolated point along bottom boundary

X_L, Y_L	coordinate description of left boundary
X_R, Y_R	coordinate description of right boundary
$X_T(s), Y_T(s)$	interpolated point along top boundary
x, y, z	physical-domain coordinates
$\alpha_1(t)$, $\alpha_2(t)$, $\alpha_3(t)$, $\alpha_4(t)$	interpolation functions for Hermite cubic connecting function
$\delta\theta$	allowed deviation of smoothed ordinate from digitized ordinate
$\Theta_1, \Theta_2, \Theta_3, \Theta_4$	parametric set descriptions of left and right boundaries
$\mu(\phi)$	cubic spline representation
ξ, η, ς	computational-domain coordinates
$\tau, \bar{\tau}$	concentration function variables
$\Phi_1, \Phi_2, \Phi_3, \Phi_4$	sets describing bottom and top boundaries in parametric form
ϕ, θ	abscissa and ordinate of points for smoothing splines
Ψ	set describing approximate arc length along cubic functions

Notation:

$\{\}_{a=b}^{a=c}$	set of points for enclosed ordered pair from initial a value b to final a value c

TABLE I. DATANEW INPUT DESCRIPTION

Description	Card image
Problem title	Maximum of 80 alphanumeric characters
Number of grid points* Bottom boundary, NB=N_B Top boundary, NT=N_T Left side boundary, NL=N_L Right side boundary, NR=N_R	NB,NT,NL,NR
x-coordinates, bottom boundary (free format)	$X_1,X_2,...,X_{NB}$
y-coordinates, bottom boundary (free format)	$Y_1,Y_2,...,Y_{NB}$
x-coordinates, top boundary (free format)	$X_1,X_2,...,X_{NT}$
y-coordinates, top boundary (free format)	$Y_1,Y_2,...,Y_{NT}$
x-coordinates, left side boundary (free format)	$X_1,X_2,...,X_{NL}$
y-coordinates, left side boundary (free format)	$Y_1,Y_2,...,Y_{NL}$
x-coordinates, right side boundary (free format)	$X_1,X_2,...,X_{NR}$
y-coordinates, right side boundary (free format)	$Y_1,Y_2,...,Y_{NR}$

*The maximum number of points currently allowed for each boundary definition is 100.

TABLE II. GENERAL OPTIONS

Number	Name	Description
1	MODIFY BOTTOM CURVE	Redefines control function $f_1(\xi)$ for bottom boundary
2	MODIFY TOP CURVE	Redefines control function $f_2(\xi)$ for top boundary
3	MODIFY SIDE CURVE	Redefines control function $f_6(\xi)$ for concentration effect of side boundaries
4	MODIFY CONNECTING CURVE	Redefines control function $f_5(\eta)$ for connecting curve
5	CHANGE KB,KT DISTRIBUTION	Controls orthogonality by assigning values to constants K_B and K_T and defining control functions $f_3(\xi)$ and $f_4(\xi)$
6	PLOT GRID DERIVATIVES	Displays a derivative along specified grid curve
7	BLOW UP A SECTION	Enlarges grid plot
8	CHANGE # OF GRID POINTS	Assigns number of grid points in each coordinate direction
9	WRITE OUT GRID AND TERMINATE	Writes current grid coordinates on file called GRIDOUT, creates file RESTART, and ends the session

TABLE III. DIGITIZING FUNCTIONS

Option*	Function
C	Corrects a previously digitized point. Place intersection of cross lines near point to be corrected and type character C. Move intersection of cross lines to desired location and again type character C. New display with correction will appear on screen.
D	Deletes a point. Place intersection of cross lines near point to be deleted and type character D. New display, minus deleted point, will appear on screen.
E	Erases all points in unit square and allows user to generate completely new curve. Type character E.
F	Completes definition of a control curve after option E has been invoked and new points have been created in unit square. For bottom, top, connecting, or side distribution functions, this character indicates to program that next and last point is (1,1). For orthogonality options, this character indicates that next and last point is $(1,y)$, where y is y-coordinate of intersection of cross lines. Type character F.
I	Digitizes point in unit square where cross lines intersect. Type character I.
L	Forces spline function to pass exactly through a point. Place intersection of cross lines at point to be digitized and type character L. Equations (6) are bypassed and deviation $\delta\theta(I)$ is set to zero. Such points are referred to as "loaded points." Type character L.
P	Replaces unit square display with a display of boundary curves. This additional display allows input of a grid point number and a digitized position on a boundary curve to generate control points in unit square. This option only applies to bottom and top boundaries. Type character P.
S	Indicates to program that digitizing process is complete or there are no changes. This option indicates to program to go to next step. Type character S.

*These commands are not followed by a CR.

TABLE IV. DATANEW FILE FOR AIRFOIL C-TYPE GRID EXAMPLE

```
AIRFOIL C-TYPE GRID EXAMPLE
81 39 2 2
1.009020  1.007465  1.002809   .995081   .984328   .970617   .954032
 .934676   .912667   .888143   .861253   .832163   .801054   .768116   .733553
 .697578   .660412   .622286   .583433   .544094   .504510   .464927   .425587
 .386735   .348608   .311442   .275467   .240904   .207967   .176857   .147768
 .120878   .096353   .074345   .054988   .038404   .024692   .013940   .006211
 .001555   .000000   .001555   .006211   .013940   .024692   .038404   .054988
 .074345   .096353   .120878   .147768   .176857   .207967   .240904   .275467
 .311442   .348608   .386735   .425587   .464927   .504510   .544094   .583433
 .622286   .660412   .697578   .733553   .768116   .801054   .832163   .861253
 .888143   .912667   .934676   .954032   .970617   .984328   .995081  1.002809
1.007465  1.009020
 .000000   .000208   .000865   .001948   .003438   .005309   .007532
 .010071   .012888   .015943   .019195   .022600   .026114   .029695   .033297
 .036874   .040379   .043762   .046972   .049954   .052653   .055014   .056980
 .058497   .059514   .059987   .059878   .059159   .057812   .055831   .053222
 .050002   .046201   .041854   .037006   .031705   .026000   .019938   .013562
 .006907   .000000  -.006907  -.013562  -.019938  -.026000  -.031705  -.037006
-.041854  -.046201  -.050002  -.053222  -.055831  -.057812  -.059159  -.059878
-.059987  -.059514  -.058497  -.056980  -.055014  -.052653  -.049954  -.046972
-.043762  -.040379  -.036874  -.033297  -.029695  -.026114  -.022600  -.019195
-.015943  -.012888  -.010071  -.007532  -.005309  -.003438  -.001948  -.000865
-.000208   .000000
1.00902    .000000  -.087156  -.173648  -.258819  -.342020  -.422618  -.500000
-.573576  -.642788  -.707107  -.766044  -.819152  -.866025  -.906308  -.939693
-.965926  -.984808  -.996195 -1.000000  -.996195  -.984808  -.965926  -.939693
-.906308  -.866025  -.819152  -.766044  -.707107  -.642788  -.573576  -.500000
-.422618  -.342020  -.258819  -.173648  -.087156   .000000 1.00902
1.000000  1.000000   .996195   .984808   .965926   .939693   .906308   .866025
 .819152   .766044   .707107   .642788   .573576   .500000   .422618   .342020
 .258819   .173648   .087156   .000000  -.087156  -.173648  -.258819  -.342020
-.422618  -.500000  -.573576  -.642788  -.707107  -.766044  -.819152  -.866025
-.906308  -.939693  -.965926  -.984808  -.996195 -1.000000 -1.00000
1.00902  1.00902
0.0 1.0
1.00902 1.00902
0.0 -1.0
```

TABLE V. DATANEW FILE FOR AIRFOIL O-TYPE GRID EXAMPLE

```
  AIRFOIL O-TYPE GRID EXAMPLE
81 91 2 2
1.009020  1.007465  1.002809   .995081   .984328   .970617   .954032   .934676
 .912667   .888143   .861253   .832163   .801054   .768116   .733553   .697578
 .660412   .622286   .583433   .544094   .504510   .464927   .425587   .386735
 .348608   .311442   .275467   .240904   .207967   .176857   .147768   .120878
 .096353   .074345   .054988   .038404   .024692   .013940   .006211   .001555
 .000000   .001555   .006211   .013940   .024692   .038404   .054988   .074345
 .096353   .120878   .147768   .176857   .207967   .240904   .275467   .311442
 .348608   .386735   .425587   .464927   .504510   .544094   .583433   .622286
 .660412   .697578   .733553   .768116   .801054   .832163   .861253   .888143
 .912667   .934676   .954032   .970617   .984328   .995081  1.002809  1.007465
1.009020
 .000000  -.000208  -.000865  -.001948  -.003438  -.005309  -.007532  -.010071
-.012888  -.015943  -.019195  -.022600  -.026114  -.029695  -.033297  -.036874
-.040379  -.043762  -.046972  -.049954  -.052653  -.055014  -.056980  -.058497
-.059514  -.059987  -.059878  -.059159  -.057812  -.055831  -.053222  -.050002
-.046201  -.041854  -.037006  -.031705  -.026000  -.019938  -.013562  -.006907
 .000000   .006907   .013562   .019938   .026000   .031705   .037006   .041854
 .046201   .050002   .053222   .055831   .057812   .059159   .059878   .059987
 .059514   .058497   .056980   .055014   .052653   .049954   .046972   .043762
 .040379   .036874   .033297   .029695   .026114   .022600   .019195   .015943
 .012888   .010071   .007532   .005309   .003438   .001948   .000865   .000208
 .000000
2.000000  1.996346  1.985402  1.967221  1.941893  1.909539  1.870318  1.824421
1.772072  1.713525  1.649067  1.579010  1.503696  1.423492  1.338789  1.250000
1.157557  1.061910   .963525   .862883   .760472   .656793   .552349   .447651
 .343207   .239528   .137117   .036475  -.061910  -.157557  -.250000  -.338789
-.423492  -.503696  -.579010  -.649067  -.713525  -.772072  -.824421  -.870318
-.909539  -.941893  -.967221  -.985402  -.996346 -1.000000  -.996346  -.985402
-.967221  -.941893  -.909539  -.870318  -.824421  -.772072  -.713525  -.649067
-.579010  -.503696  -.423492  -.338789  -.250000  -.157557  -.061910   .036475
 .137117   .239528   .343207   .447651   .552349   .656793   .760472   .862883
 .963525  1.061910  1.157557  1.250000  1.338789  1.423492  1.503696  1.579010
1.649067  1.713525  1.772072  1.824421  1.870318  1.909539  1.941893  1.967221
1.985402  1.996346  2.000000
 .000000  -.104635  -.208760  -.311868  -.413456  -.513030  -.610105  -.704207
-.794879  -.881678  -.964181 -1.041988 -1.114717 -1.182016 -1.243556 -1.299038
-1.348191 -1.390776 -1.426585 -1.455444 -1.477212 -1.491783 -1.499086 -1.49908
-1.491783 -1.477212 -1.455444 -1.426585 -1.390776 -1.348191 -1.299038 -1.24355
-1.182016 -1.114717 -1.041988  -.964181  -.881678  -.794879  -.704207  -.61010
-.513030  -.413456  -.311868  -.208760  -.104635   .000000   .104635   .208760
 .311868   .413456   .513030   .610105   .704207   .794879   .881678   .964181
1.041988  1.114717  1.182016  1.243556  1.299038  1.348191  1.390776  1.426585
1.455444  1.477212  1.491783  1.499086  1.499086  1.491783  1.477212  1.455444
1.426585  1.390776  1.348191  1.299038  1.243556  1.182016  1.114717  1.041988
 .964181   .881678   .794879   .704207   .610105   .513030   .413456   .311868
 .208760   .104635   .000000
1.009020  2.000000
 .000000   .000000
1.009020  2.000000
 .000000   .000000
```

TABLE VI. DATANEW FILE FOR NOZZLE CONFIGURATION MINUS SIDE BOUNDARIES

```
 NOZZLE CONFIGURATION MINUS SIDE BOUNDARIES
 50 50 0 0
-13.00000  -12.46939  -11.93878  -11.40816  -10.87755
-10.34694   -9.81633   -9.28571   -8.75510   -8.22449
 -7.69388   -7.16327   -6.63265   -6.10204   -5.57143
 -5.04082   -4.51020   -3.97959   -3.44898   -2.91837
 -2.38776   -1.85714   -1.32653    -.79592    -.26531
   .26531     .79592    1.32653    1.85714    2.38776
  2.91837    3.44898    3.97959    4.51020    5.04082
  5.57143    6.10204    6.63265    7.16327    7.69388
  8.22449    8.75510    9.28571    9.81633   10.34694
 10.87755   11.40816   11.93878   12.46939   13.00000
 -7.00000   -6.73806   -6.47711   -6.21726   -5.95866
 -5.70148   -5.44592   -5.19222   -4.94067   -4.69161
 -4.44547   -4.20274   -3.96407   -3.73023   -3.50219
 -3.28115   -3.06863   -2.86653   -2.67721   -2.50356
 -2.34907   -2.21774   -2.11390   -2.04173   -2.00468
 -2.00468   -2.04173   -2.11390   -2.21774   -2.34907
 -2.50356   -2.67721   -2.86653   -3.06863   -3.28115
 -3.50219   -3.73023   -3.96407   -4.20274   -4.44547
 -4.69161   -4.94067   -5.19222   -5.44592   -5.70148
 -5.95866   -6.21726   -6.47711   -6.73806   -7.00000
-13.00000  -12.46939  -11.93878  -11.40816  -10.87755
-10.34694   -9.81633   -9.28571   -8.75510   -8.22449
 -7.69388   -7.16327   -6.63265   -6.10204   -5.57143
 -5.04082   -4.51020   -3.97959   -3.44898   -2.91837
 -2.38776   -1.85714   -1.32653    -.79592    -.26531
   .26531     .79592    1.32653    1.85714    2.38776
  2.91837    3.44898    3.97959    4.51020    5.04082
  5.57143    6.10204    6.63265    7.16327    7.69388
  8.22449    8.75510    9.28571    9.81633   10.34694
 10.87755   11.40816   11.93878   12.46939   13.00000
  7.00000    6.73806    6.47711    6.21726    5.95866
  5.70148    5.44592    5.19222    4.94067    4.69161
  4.44547    4.20274    3.96407    3.73023    3.50219
  3.28115    3.06863    2.86653    2.67721    2.50356
  2.34907    2.21774    2.11390    2.04173    2.00468
  2.00468    2.04173    2.11390    2.21774    2.34907
  2.50356    2.67721    2.86653    3.06863    3.28115
  3.50219    3.73023    3.96407    4.20274    4.44547
  4.69161    4.94067    5.19222    5.44592    5.70148
  5.95866    6.21726    6.47711    6.73806    7.00000
```

TABLE VII. DATANEW FILE FOR NOZZLE CONFIGURATION WITH SIDE BOUNDARIES

```
 NOZZLE CONFIGURATION WITH SIDE BOUNDARIES
 50 50 21 21
-13.00000  -12.46939  -11.93878  -11.40816  -10.87755
-10.34694   -9.81633   -9.28571   -8.75510   -8.22449
 -7.69388   -7.16327   -6.63265   -6.10204   -5.57143
 -5.04082   -4.51020   -3.97959   -3.44898   -2.91837
 -2.38776   -1.85714   -1.32653    -.79592    -.26531
   .26531     .79592    1.32653    1.85714    2.38776
  2.91837    3.44898    3.97959    4.51020    5.04082
  5.57143    6.10204    6.63265    7.16327    7.69388
  8.22449    8.75510    9.28571    9.81633   10.34694
 10.87755   11.40816   11.93878   12.46939   13.00000
 -7.00000   -6.73806   -6.47711   -6.21726   -5.95866
 -5.70148   -5.44592   -5.19222   -4.94067   -4.69161
 -4.44547   -4.20274   -3.96407   -3.73023   -3.50219
 -3.28115   -3.06863   -2.86653   -2.67721   -2.50356
 -2.34907   -2.21774   -2.11390   -2.04173   -2.00468
 -2.00468   -2.04173   -2.11390   -2.21774   -2.34907
 -2.50356   -2.67721   -2.86653   -3.06863   -3.28115
 -3.50219   -3.73023   -3.96407   -4.20274   -4.44547
 -4.69161   -4.94067   -5.19222   -5.44592   -5.70148
 -5.95866   -6.21726   -6.47711   -6.73806   -7.00000
-13.00000  -12.46939  -11.93878  -11.40816  -10.87755
-10.34694   -9.81633   -9.28571   -8.75510   -8.22449
 -7.69388   -7.16327   -6.63265   -6.10204   -5.57143
 -5.04082   -4.51020   -3.97959   -3.44898   -2.91837
 -2.38776   -1.85714   -1.32653    -.79592    -.26531
   .26531     .79592    1.32653    1.85714    2.38776
  2.91837    3.44898    3.97959    4.51020    5.04082
  5.57143    6.10204    6.63265    7.16327    7.69388
  8.22449    8.75510    9.28571    9.81633   10.34694
 10.87755   11.40816   11.93878   12.46939   13.00000
  7.00000    6.73806    6.47711    6.21726    5.95866
  5.70148    5.44592    5.19222    4.94067    4.69161
  4.44547    4.20274    3.96407    3.73023    3.50219
  3.28115    3.06863    2.86653    2.67721    2.50356
  2.34907    2.21774    2.11390    2.04173    2.00468
  2.00468    2.04173    2.11390    2.21774    2.34907
  2.50356    2.67721    2.86653    3.06863    3.28115
  3.50219    3.73023    3.96407    4.20274    4.44547
  4.69161    4.94067    5.19222    5.44592    5.70148
  5.95866    6.21726    6.47711    6.73806    7.00000
-13.0 -12.073 -11.2366 -10.573 -10.1467
-10.0 -10.1467 -10.573 -11.2366 -12.073 -13.0
-13.927 -14.7625 -15.427 -15.8533 -16.0
-15.8533 -15.427 -14.7625 -13.927 -13.0
-7.0 -6.29999 -5.59999 -4.89999 -4.19999
-3.49999 -2.80001 -2.10001 -1.39996 -0.7001
0.0 0.7001 1.39996 2.10001 2.80001 3.49999
4.19999 4.89999 5.59999 6.29999 7.0
 13.00000   12.45946   11.95849   11.50226   11.09628
 10.74625   10.45780   10.23613   10.08564   10.00955 10.0
 10.00955   10.08564   10.23613   10.45780   10.74625
 11.09628   11.50226   11.95849   12.45946   13.00000
 -7.00000   -6.26316   -5.52632   -4.78947   -4.05263
 -3.31579   -2.57895   -1.84211   -1.10526    -.36842 0.0
   .36842    1.10526    1.84211    2.57895    3.31579
  4.05263    4.78947    5.52632    6.26316    7.00000
```

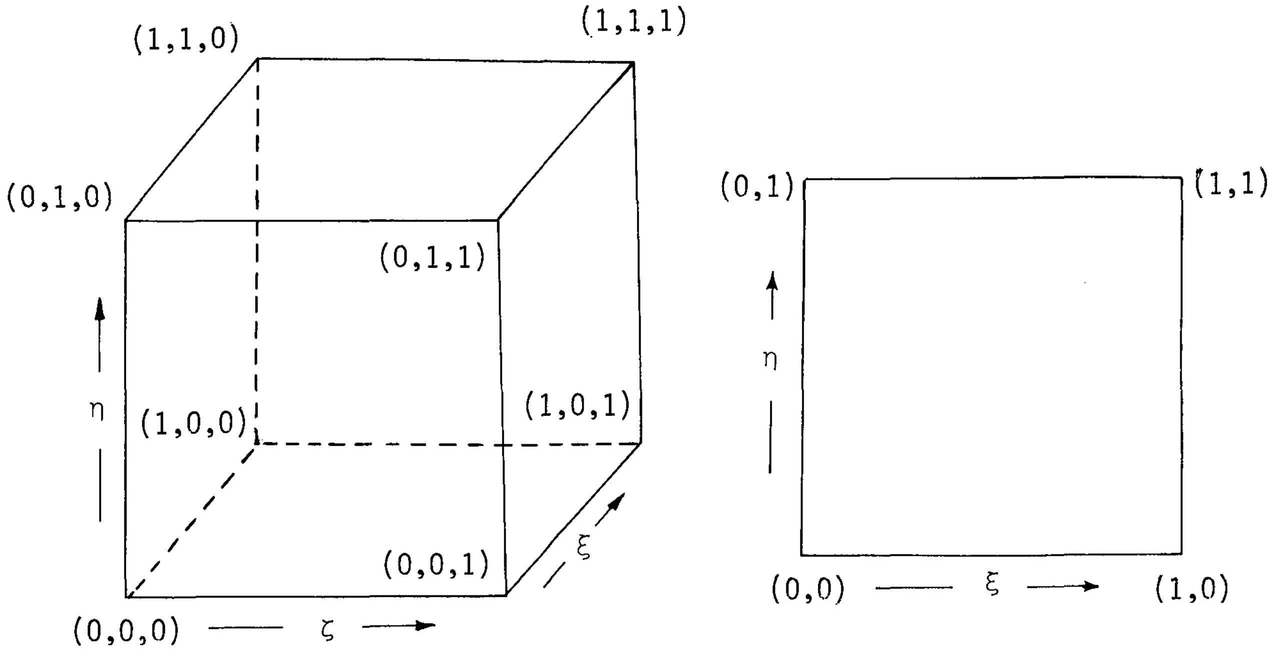


Figure 1. Computational domains.

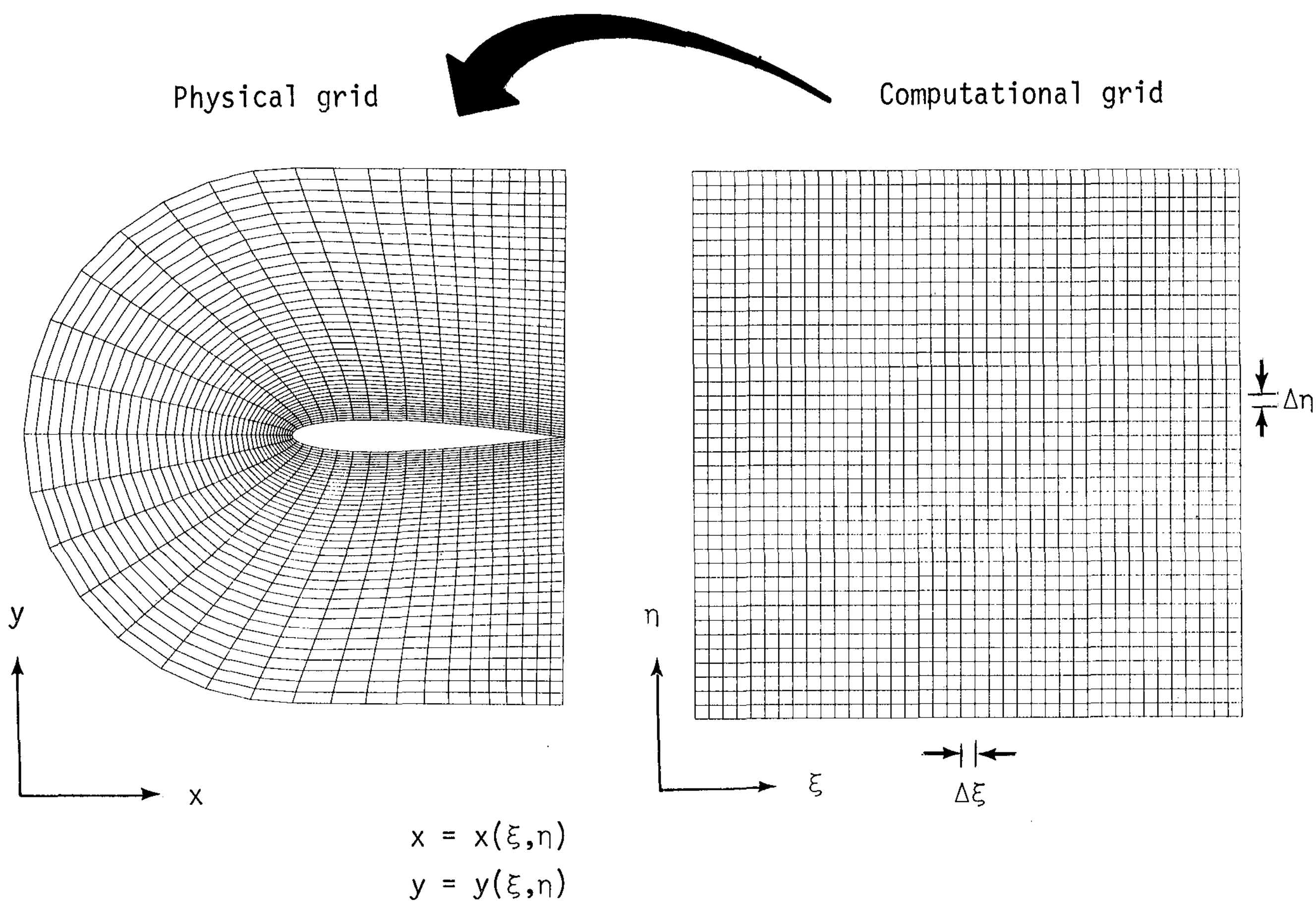


Figure 2. Physical grid versus computational grid.

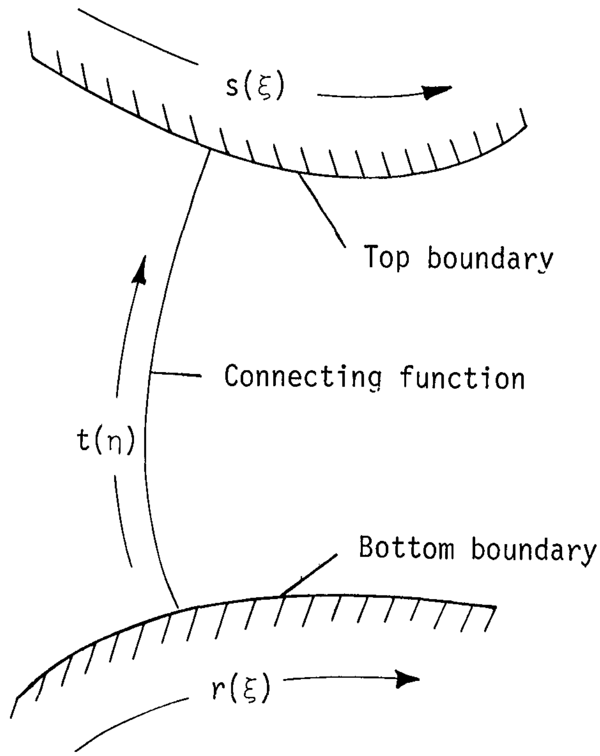


Figure 3. Curved connecting function.

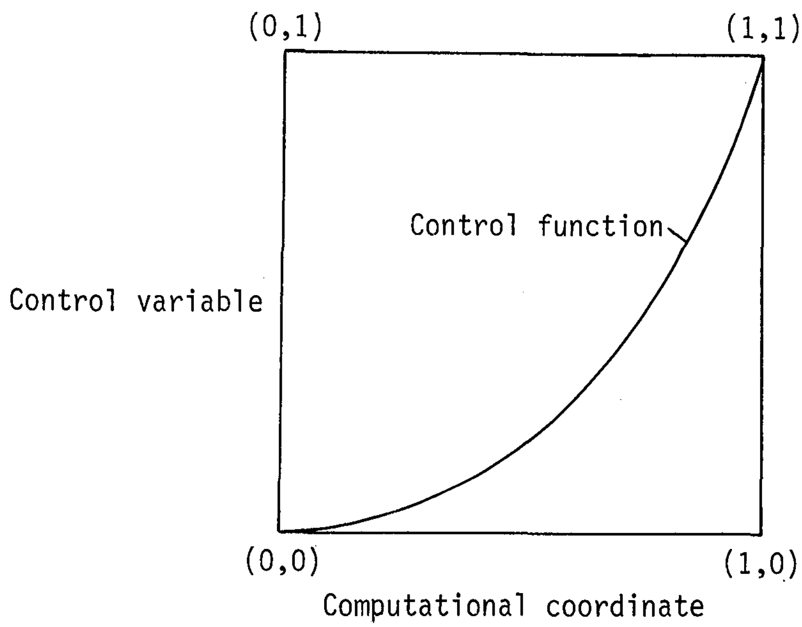


Figure 4. Control domain.

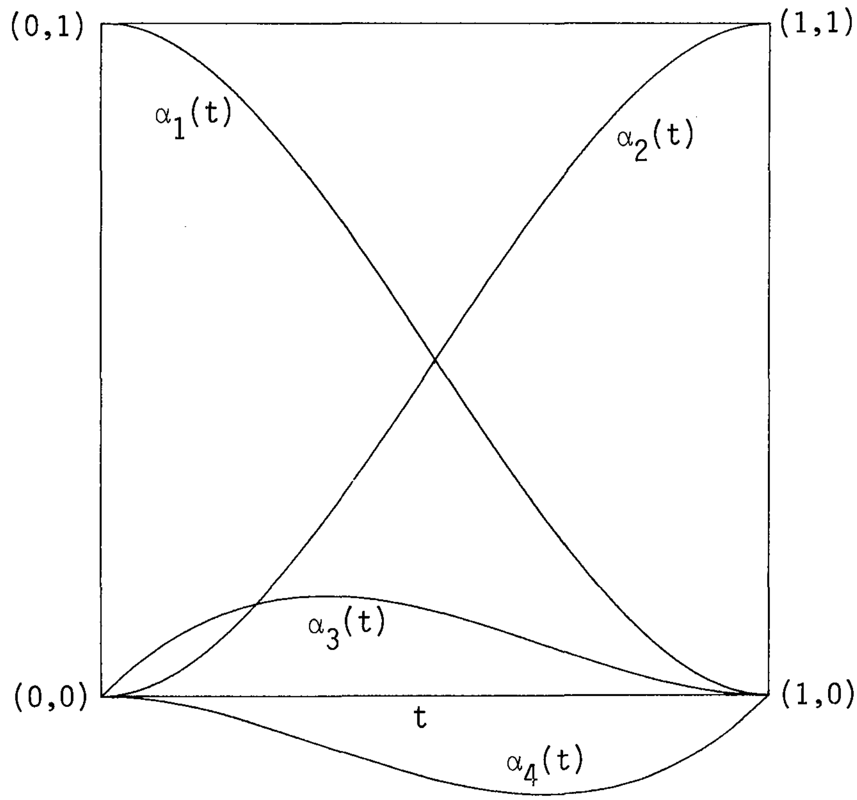


Figure 5. Cubic blending functions.

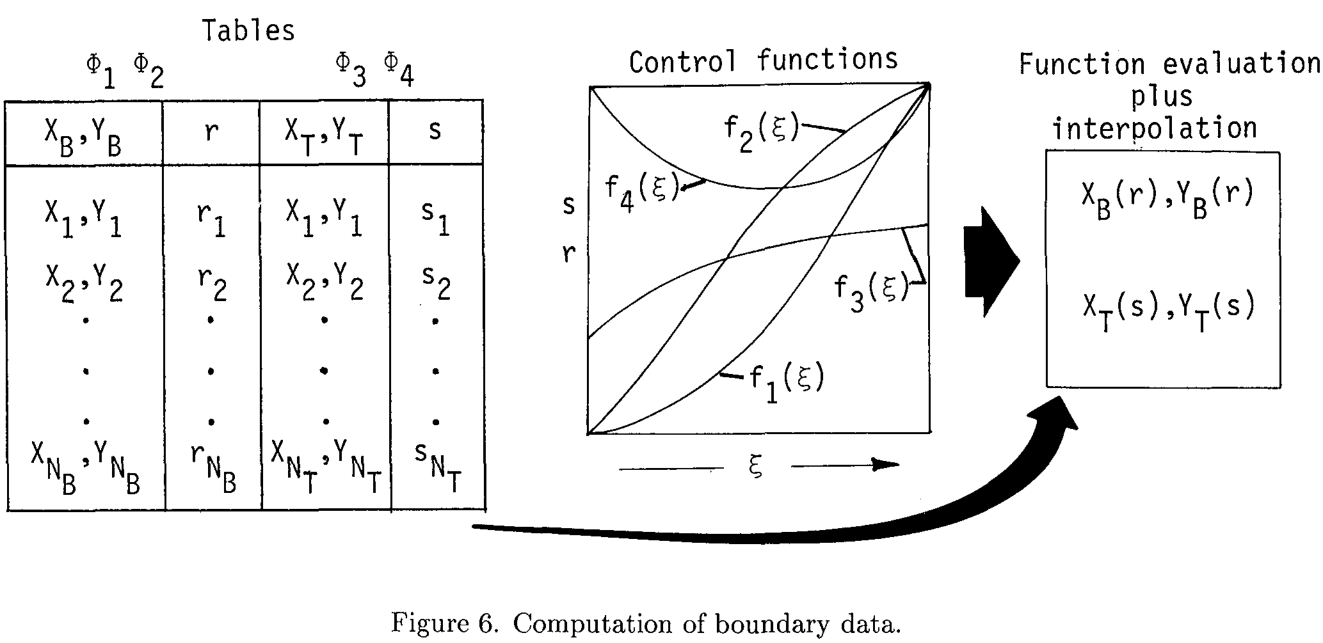


Figure 6. Computation of boundary data.

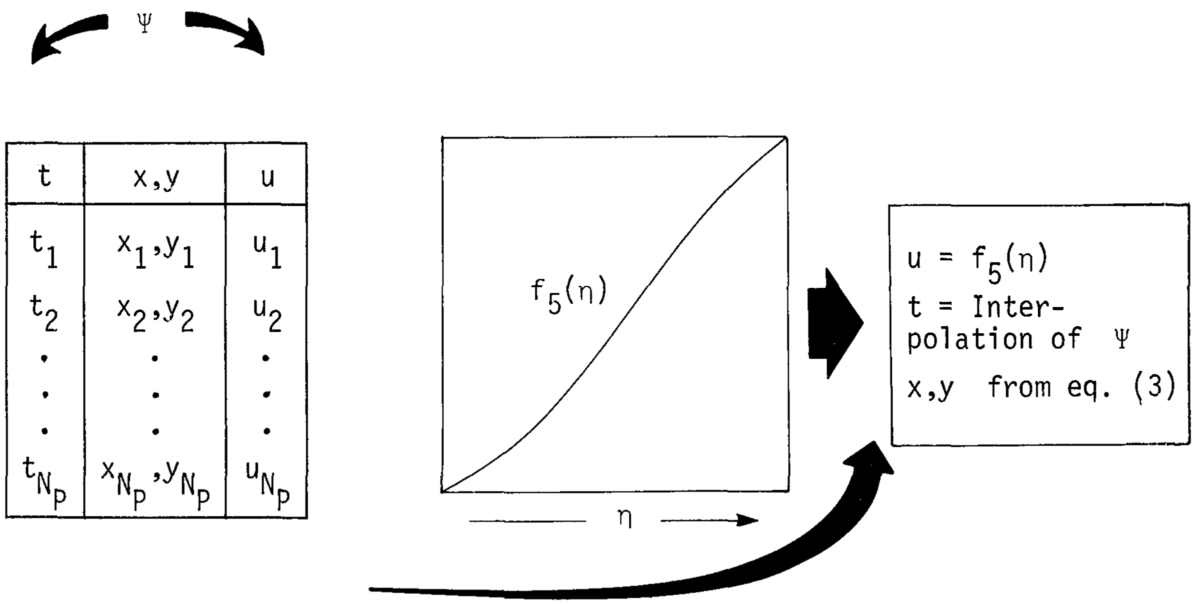


Figure 7. Computation of points along connecting function.

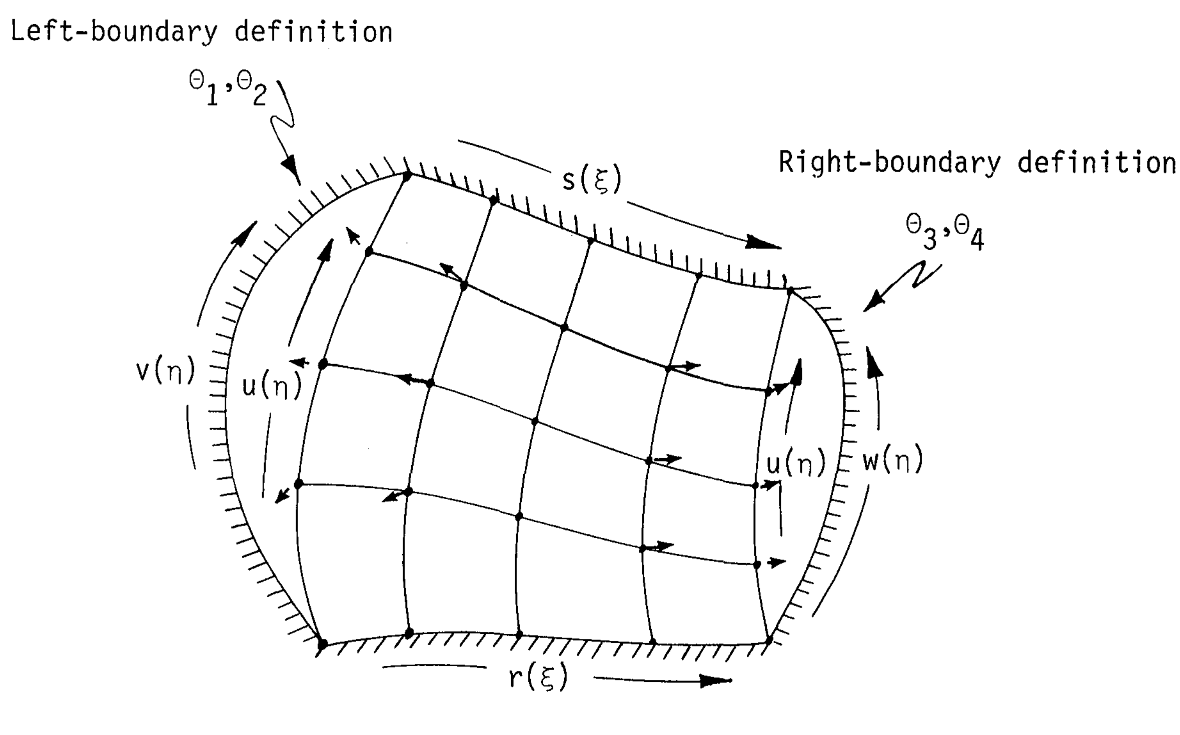


Figure 8. Left and right boundaries.

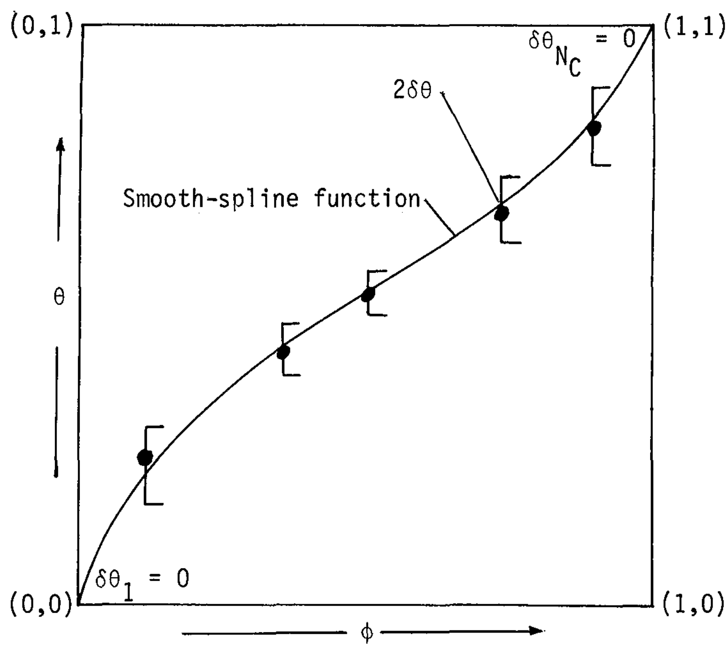


Figure 9. Smooth-spline control functions.

OPTIONS

1. MODIFY BOTTOM CURVE
2. MODIFY TOP CURVE
3. MODIFY SIDE CURVE
4. MODIFY CONNECTING CURVE
5. CHANGE KB,KT DISTRIBUTION
6. PLOT GRID DERIVATIVES
7. BLOW UP A SECTION
8. CHANGE # OF GRID POINTS
9. WRITE OUT GRID AND TERMINATE

ENTER INTEGER VALUE AND PRESS "RETURN"

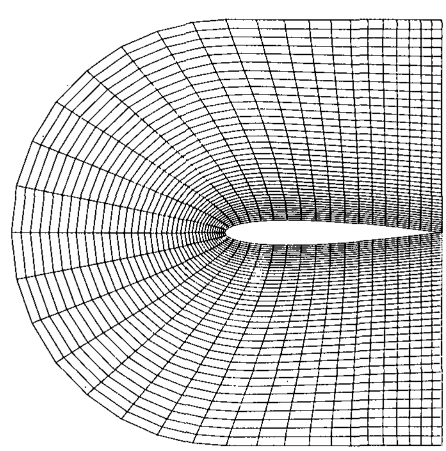


Figure 10. General options and grid display.

OPTIONS

ENTER INTEGER VALUE FOR TYPE OF NOMINAL CURVE

1 LINEAR

2 SINGLE EXPONENTIAL

3 BI–EXPONENTIAL

4 USER DEFINED

5 EQUATE TO TOP CURVE (FOR BOTTOM CURVE)

Figure 11. Local option display for grid spacing control.

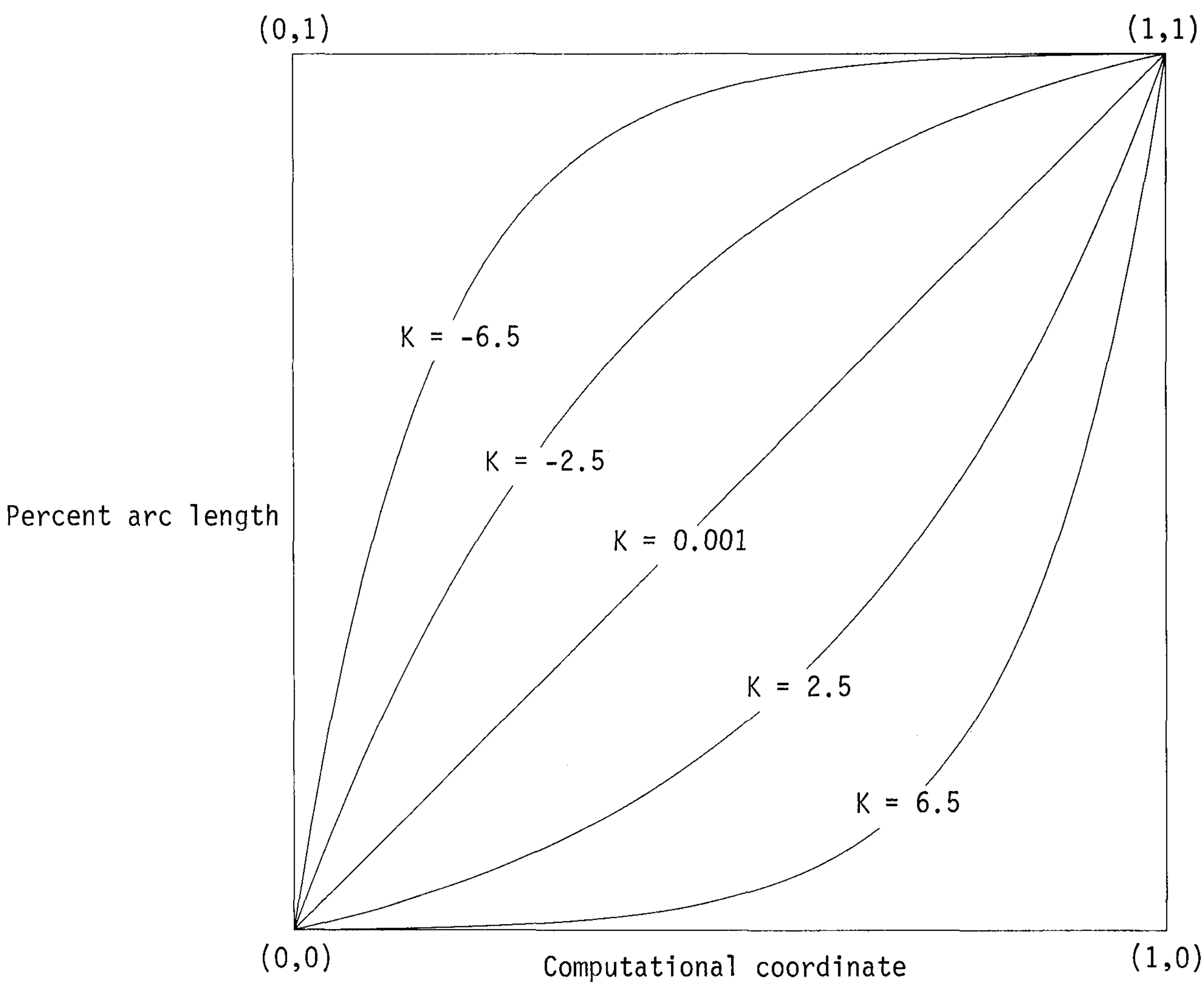


Figure 12. Single exponential function for grid spacing control.

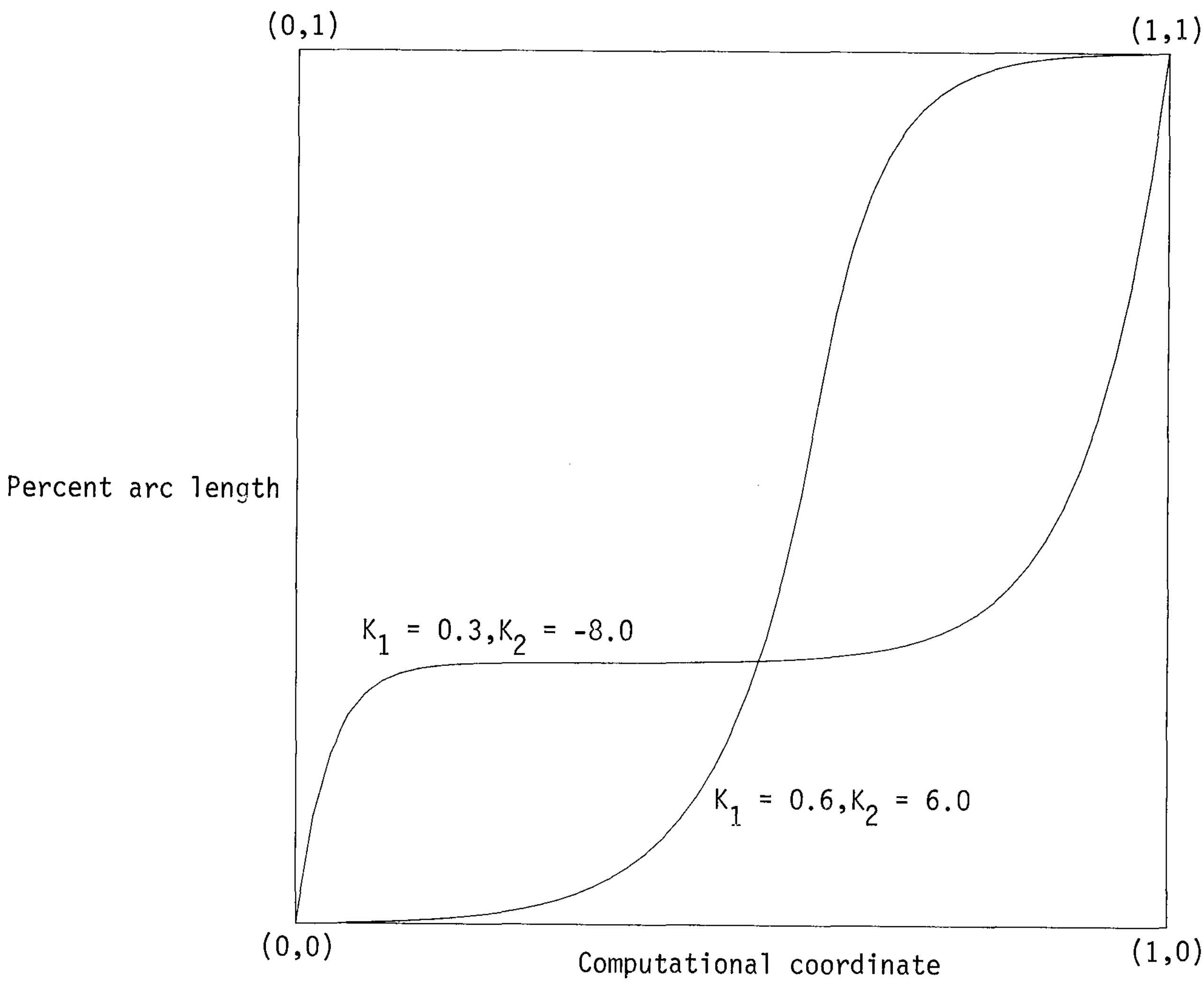


Figure 13. Biexponential function (local option 3) for grid spacing control.

OPTIONS FUNCTIONS

OPTIONS	FUNCTIONS
C	CORRECT A PREVIOUSLY DIGITIZED POINT
D	DELETE A POINT
E	ERASE ALL POINTS
F	COMPLETE THE DEFINITION OF A CONTROL CURVE
I	DIGITIZE A POINT IN THE UNIT SQUARE
L	LOAD A POINT (NO SMOOTHING)
P	PRESENT BOUNDARY CURVE DISPLAY
S	COMPLETE DIGITIZING PROCESS

TYPE LETTER FOR DESIRED FUNCTION

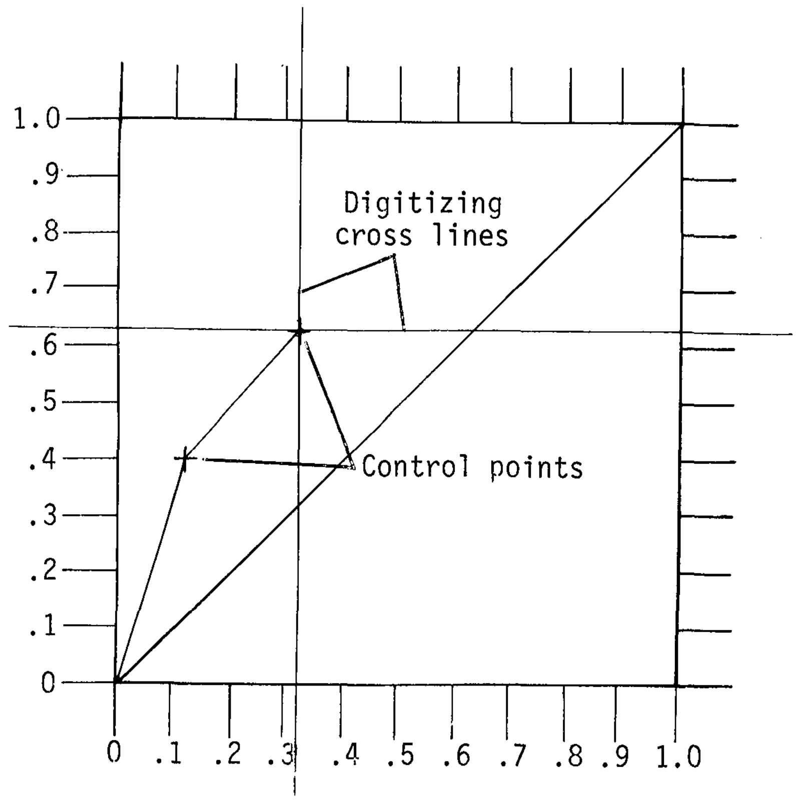


Figure 14. Display for creating arbitrary control functions.

THE BOUNDARY BEING WORKED ON IS A SOLID LINE

GRID LINE #1 IS MARKED BY A DIAMOND

LOADED POINTS ARE SHOWN BY A "+" SIGN

A MAXIMUM OF 10 POINTS CAN BE LOADED

CURRENTLY 1 POINT IS LOADED

THE LOAD OPTIONS ARE:

1: SHIFT AN ORIGINAL GRID LINE TO A NEW LOCATION, RANGE 2 THRU 10

2: CHANGE A SHIFTED GRID LINE POSITION

ENTER THE GRID LINE NUMBER (ENTER 0 TO STOP) AND PRESS "RETURN"

LETTER	FUNCTION
C	TO CHANGE A SHIFTED GRID LINE
I	TO SHIFT A GRID LINE
S	TO STOP LOADING MODE

MOVE CURSOR TO
POSITION ON BOUNDARY
THEN TYPE LETTER
FOR DESIRED FUNCTION

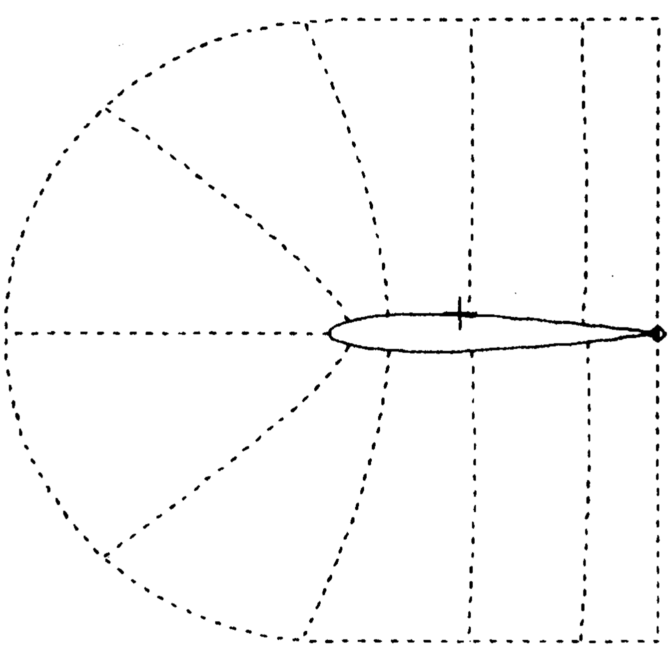


Figure 15. Display to aid in control of grid spacing along boundaries.

OPTIONS

1 CHANGE THE SMOOTHING OF THE CURVE

2 CHANGE THE CURVE

3 NO CHANGE

ENTER INTEGER VALUE AND PRESS "RETURN"

"1" CURVE SMOOTHING OPTION

THE VARIABLES M0 AND M1 ENABLE THE USER TO SMOOTH THE CURVE GENERATED BY THE PROGRAM

THE CURRENT VALUES OF M0 AND M1 ARE .010, .010

TYPE IN THE VALUES SEPARATED BY A COMMA

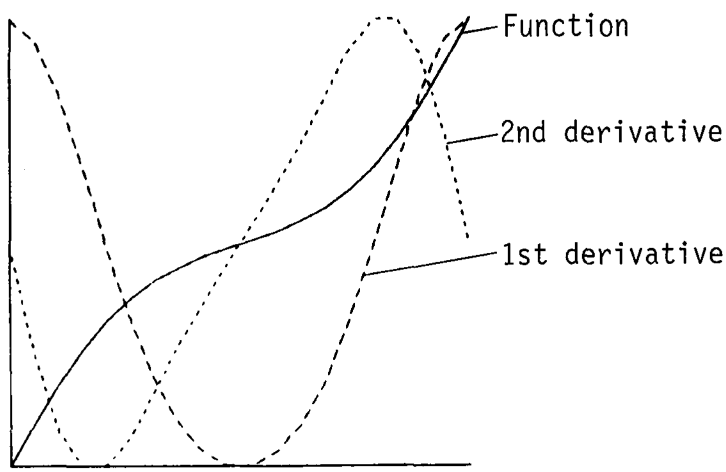


Figure 16. Display of smooth-grid-spacing control function and local options.

SELECT MAGNITUDES OF ORTHOGONALITY OF GRID

KB (ABSOLUTE RANGE 0.0 TO 30.0)

KT (ABSOLUTE RANGE 0.0 TO 15.0)

PRESENT VALUES ARE .000, .000

ENTER TWO REAL NUMBERS SEPARATED BY A COMMA AND PRESS "RETURN"

SELECT KB,KT DISTRIBUTIONS; OPTIONS ARE:

0 CONSTANT DISTRIBUTION

1 USER–DEFINED DISTRIBUTION

2 KEEP CURRENT DISTRIBUTION

CURRENT VALUES ARE 0,0

ENTER TWO INTEGER VALUES SEPARATED BY A COMMA AND PRESS "RETURN"

Figure 17. Display for grid orthogonality (general option 5).

SELECT LINE TYPE, LINE NUMBER, DERIVATIVE TYPE, AND DIRECTION
LINE TYPE 1 ALONG BOUNDARY
2 ACROSS BOUNDARIES
DERIVATIVE TYPE 1 FOR X
2 FOR Y
DERIVATIVE DIRECTION 1 ALONG BOUNDARY
2 ACROSS BOUNDARIES
ENTER 0,0,0,0 TO QUIT THIS MODE
ENTER FOUR INTEGER VALUES SEPARATED BY COMMAS AND PRESS "RETURN"

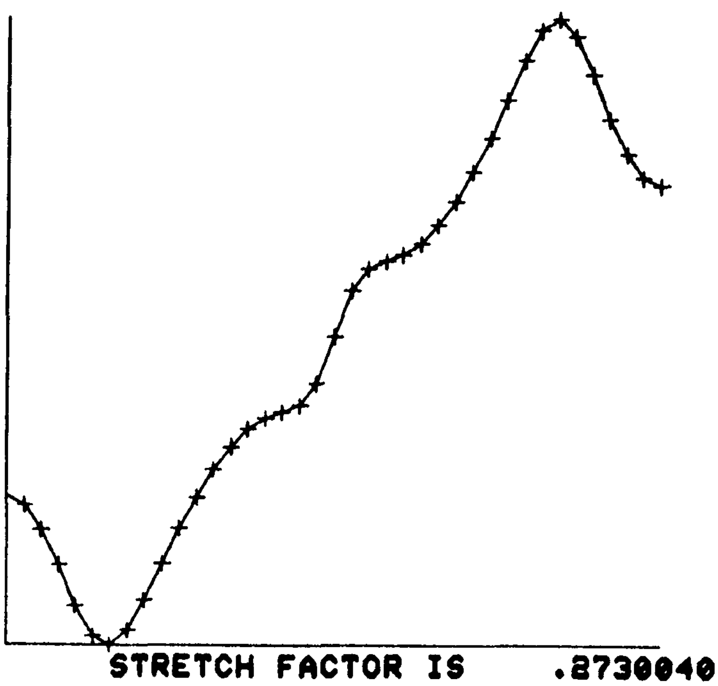


Figure 18. Display for grid derivatives.

SELECT THE MULTIPLE TO CHANGE THE SIZE OF THE GRID BY
(1.0 TO RESTORE THE ORIGINAL GRID, 0.0 TO EXIT MODE)
ENTER REAL VALUE AND PRESS "RETURN"

PLACE CURSOR AT THE CENTER OF THE REGION TO BE BLOWN UP AND PRESS THE LETTER "I"

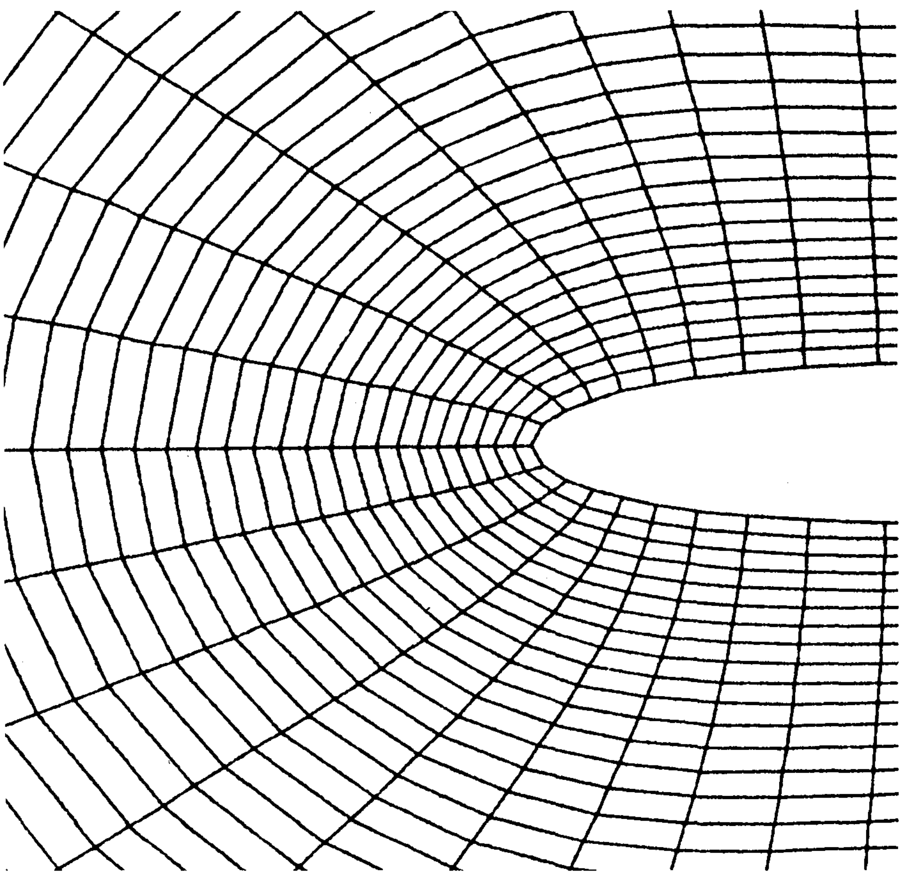


Figure 19. Display for grid enlargement (general option 7).

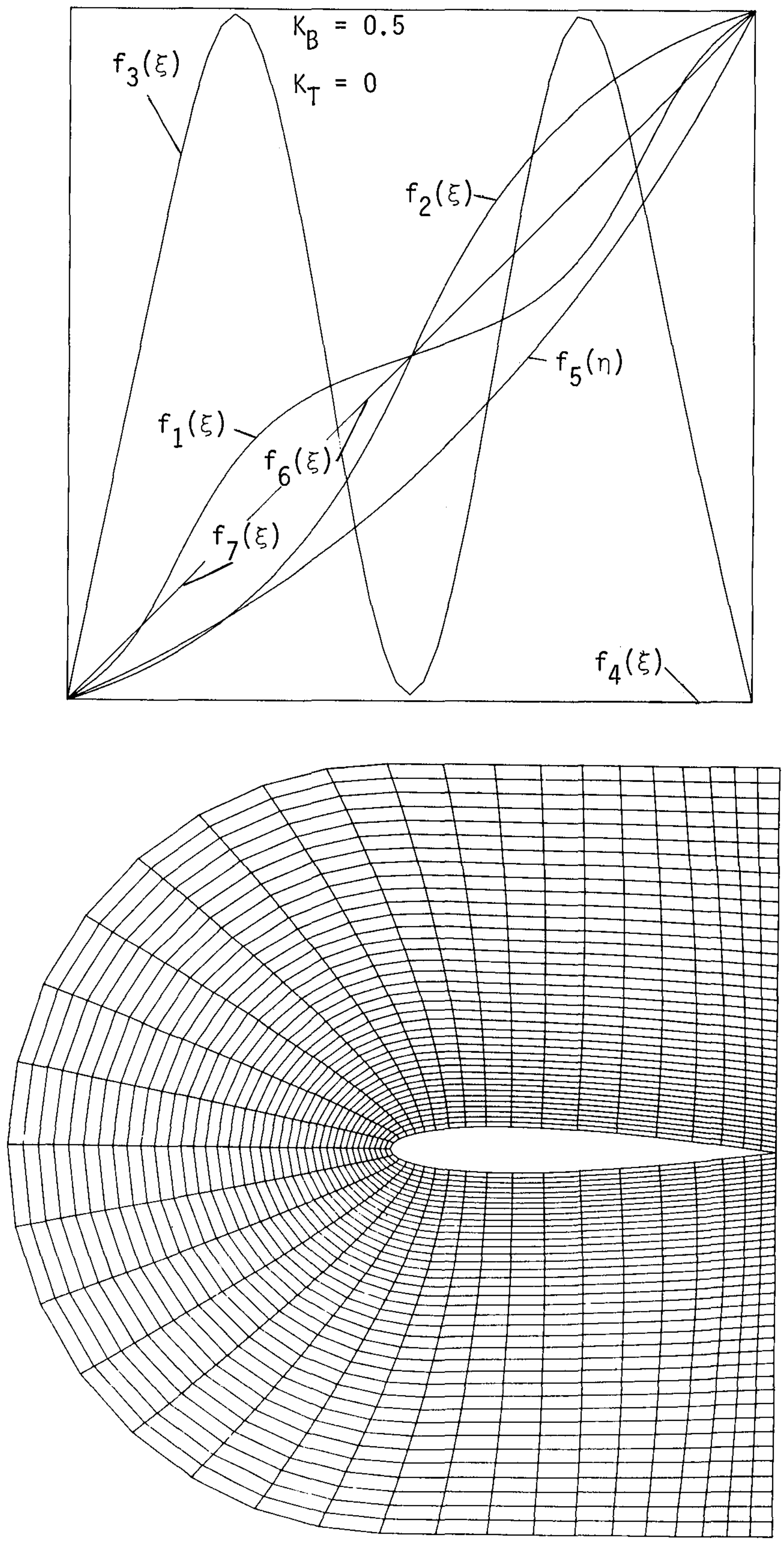


Figure 20. Control functions and C-type grid for an airfoil.

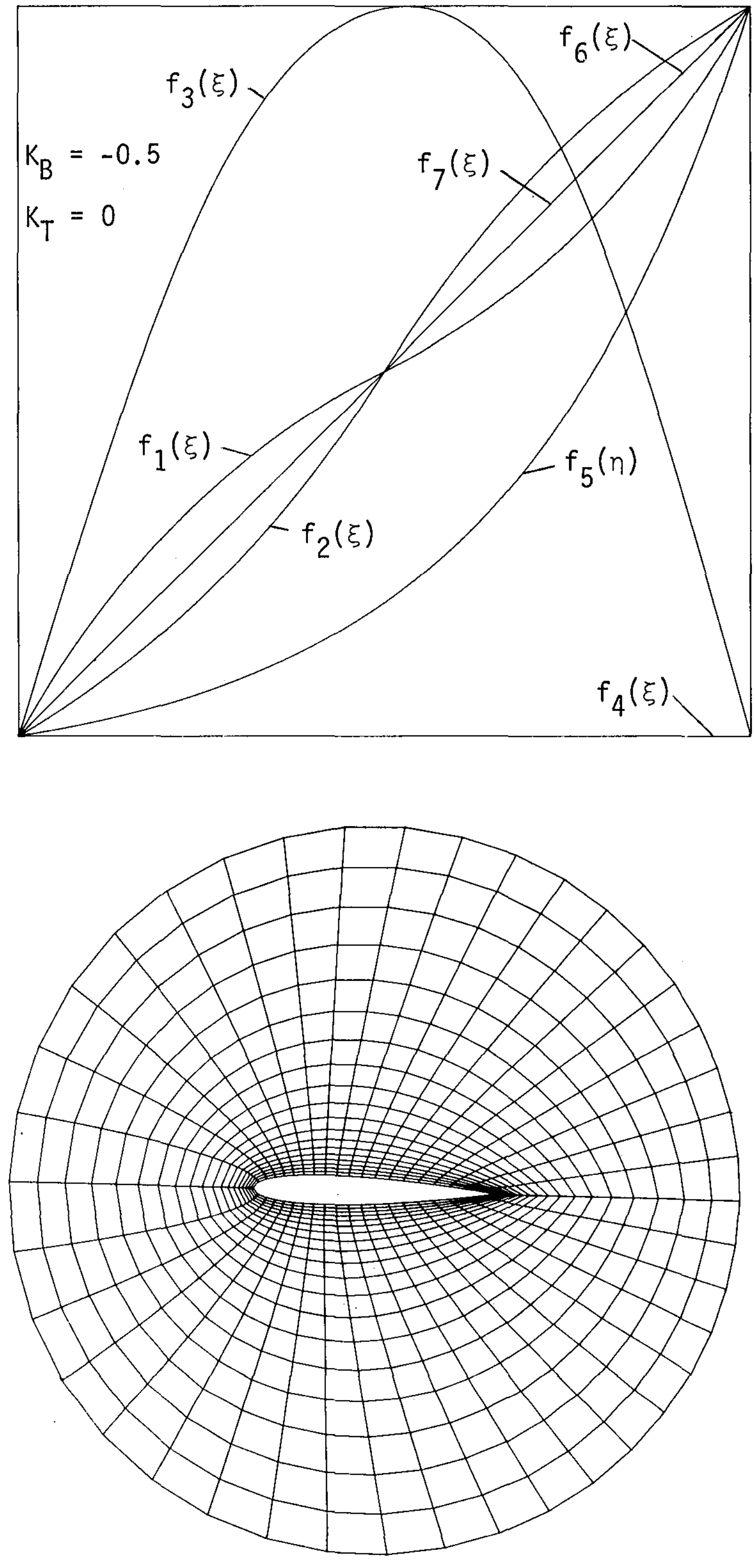


Figure 21. Control functions and O-type grid for an airfoil.

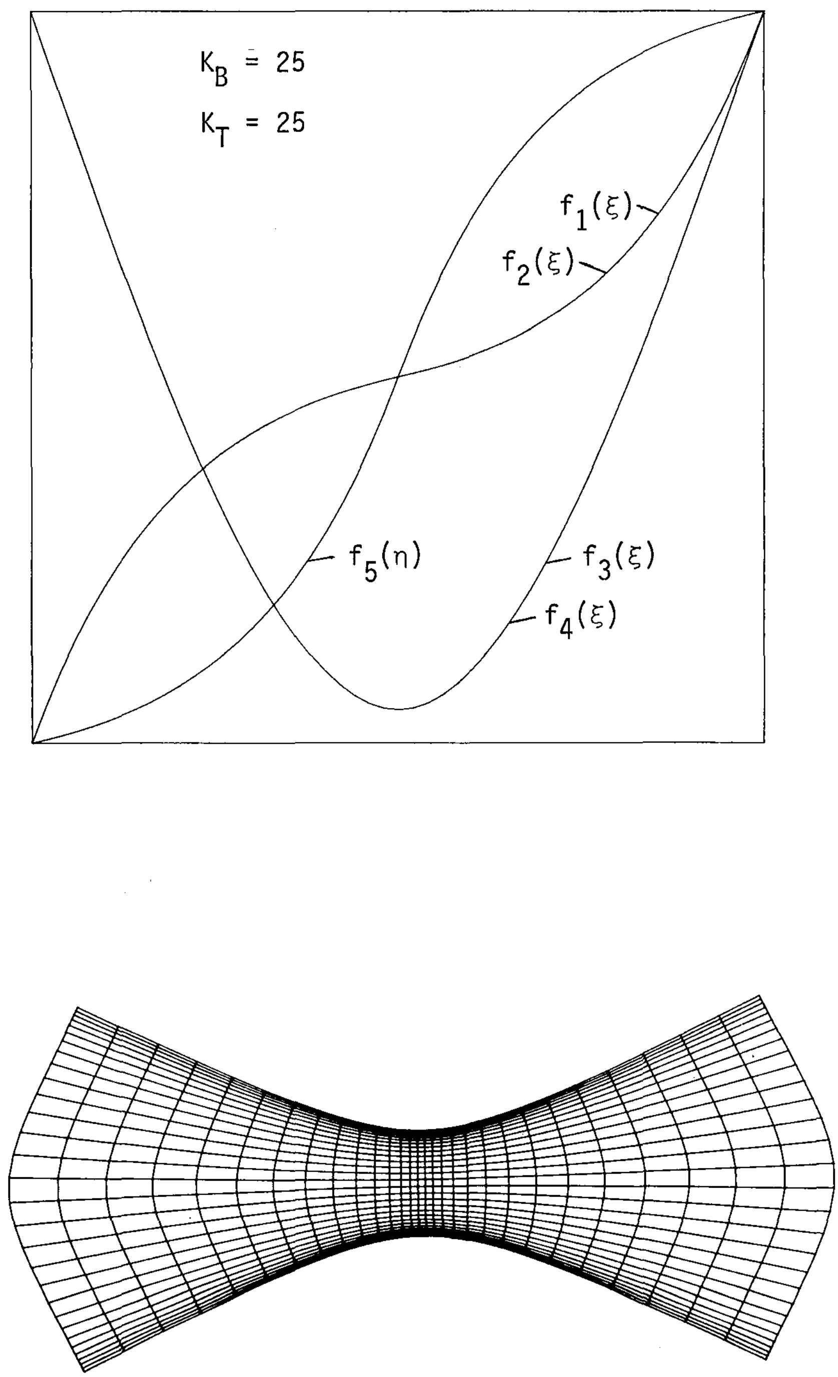
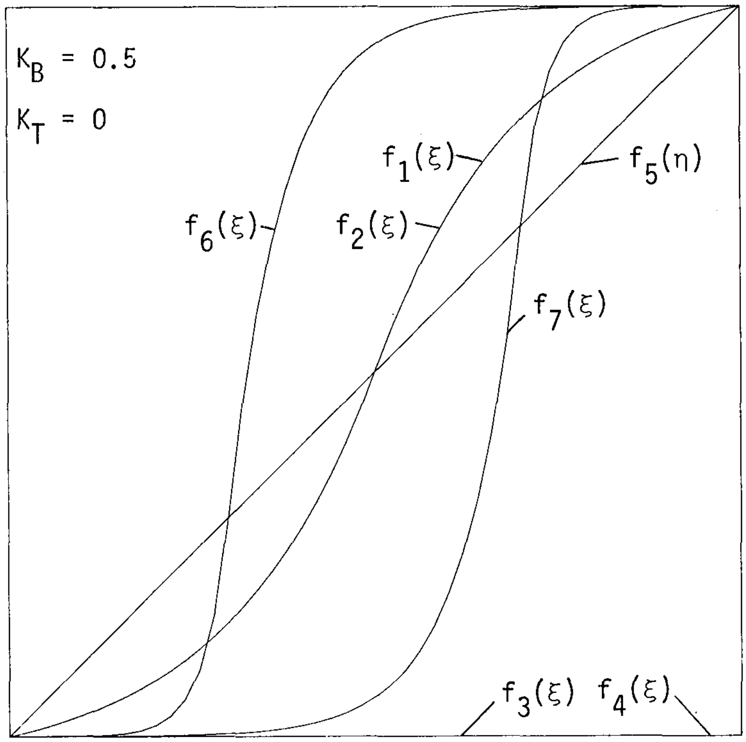


Figure 22. Control functions and grid for a nozzle configuration.



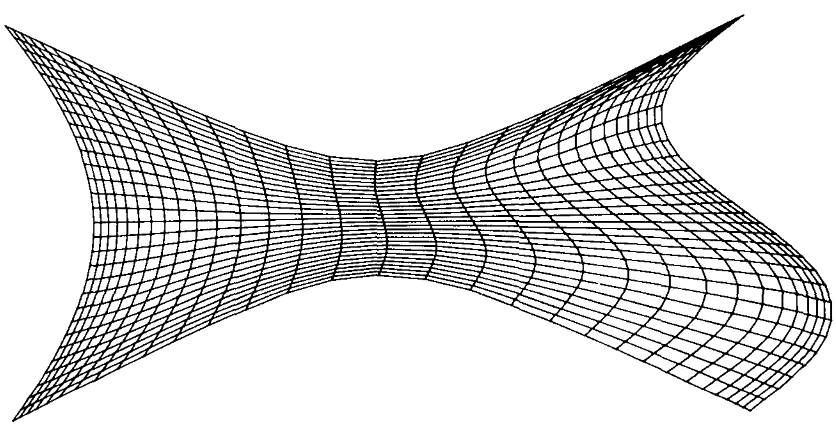


Figure 23. Control functions and grid for nozzle configuration with side boundaries specified.

1. Report No. NASA TP-2533	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle Interactive Algebraic Grid-Generation Technique		5. Report Date March 1986
		6. Performing Organization Code 505-31-83-02
7. Author(s) R. E. Smith and M. R. Wiese		8. Performing Organization Report No. L-16019
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225		10. Work Unit No.
		11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001		13. Type of Report and Period Covered Technical Paper
		14. Sponsoring Agency Code

15. Supplementary Notes
R. E. Smith: NASA Langley Research Center, Hampton, Virginia.
M. R. Wiese: Computer Sciences Corporation, Hampton, Virginia.

16. Abstract
An algebraic grid-generation technique and use of an associated interactive computer program are described. The technique, called the two-boundary technique, is based on Hermite cubic interpolation between two fixed, nonintersecting boundaries. The boundaries are referred to as the bottom and top, and they are defined by two ordered sets of points. Left and right side boundaries which intersect the bottom and top boundaries may also be specified by two ordered sets of points. When side boundaries are specified, linear blending functions are used to conform interior interpolation to the side boundaries. Spacing between physical-grid coordinates is determined as a function of boundary data and uniformly spaced computational coordinates. Control functions relating computational coordinates to parametric intermediate variables that affect the distance between grid points are embedded in the interpolation formulas. A versatile control function technique with smooth-cubic-spline functions is presented. The technique works best in an interactive graphics environment where computational displays and user responses are quickly exchanged. An interactive computer program based on the technique and called TBGG (two-boundary grid generation) is also described.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Computational fluid dynamics Grid generation Interactive graphics	Unclassified—Unlimited Subject Category 59

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	38	A03

For sale by the National Technical Information Service, Springfield, Virginia 22161

NASA-Langley, 1986

National Aeronautics and
Space Administration
Code NIT-4

Washington, D.C.
20546-0001

Official Business
Penalty for Private Use, $300

BULK RATE
POSTAGE & FEES PAID
NASA
Permit No. G-27

NASA

POSTMASTER: If Undeliverable (Section 158
Postal Manual) Do Not Return